MCU Driver Implementation for AFE4460

September, 2022

Features:

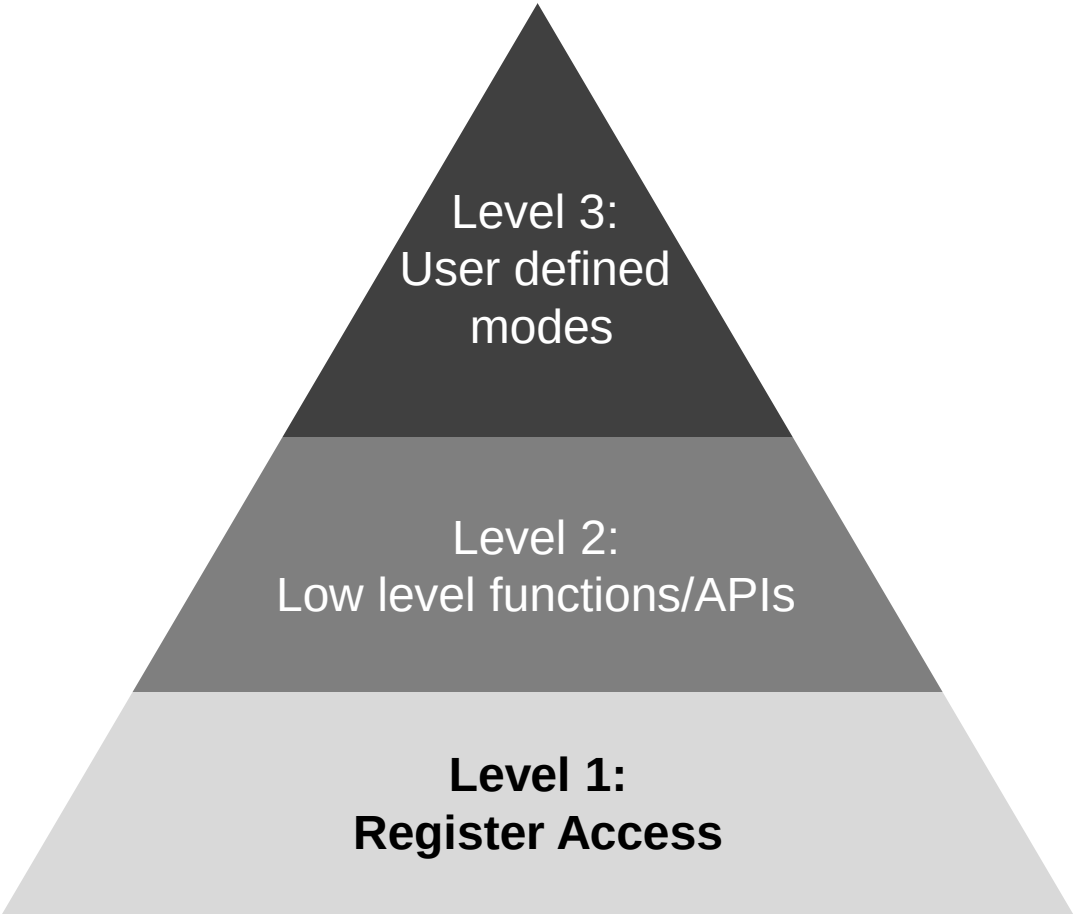
- **Written in C** : MCU independent.
- **Multi level abstraction**: Low level register access, functions for block level configurations.
- **Built-in error checks**: Error free AFE configurations.
- Optimized and tested functions for best performance.
- Example configurations based on TI’s platform with optimum settings.
- Easy referencing with the datasheet as exact names are used.

Driver Framework: Abstraction

- User can call multiple low level functions to configure the AFE in particular mode.
- There are few example configurations such as HRM, SpO2 based on AFE4460EVM. User can use these as reference and modify based on their hardware platform.
  E.g. **AFE_configAFE_forHRM**()
- If needed, user can modify these given examples to change any AFE parameter such as RF, LED_ON time or write a completely new mode with the help of several low level functions.

- **AFE_Functions_PPG (*.c and *.h file)**
- Each function sets one or multiple AFE parameters based on conditions.
- E.g : **AFE_config_clockMode** (CLK_MODE_EXT);

- **AFE_RegMap (*.c and *.h file)**
- Contains definitions of all AFE parameters.
- AFE Parameters can be accessed using:
  - **AFE_modifyRegGlobal** : For Global Registers
  - **AFE_modifyRegPPM** : For Per Phase Registers



Available Low Level Functions:

Global Configurations

- SW RST
- SW power down
- Clocking Scheme
- Phase Timing Scheme
- Enable / Disable timing engine
- Full scale for LED Current (i.e. ILED_FS)
- Bandwidth for both Noise Reduction Filters.
- Full scale for AMB_DAC
- Re-convergence threshold for LED DC Cancellation
- Scaling factor for DRE
- Sampling frequency
- Interrupt on ADC_RDY pin

Per-Phase Configurations

- LED to be used (TXN and TXP) for both LED Driver
- LED ON time
- LED current for each LED driver
- NUMAV
- Auto insertion of AMBs
- Filter BW selection
- Ambient Cancellation scheme
- LED DC Cancellation scheme
- FIFO Data control
- Controls for each TIA
  - PD to be used
  - RF and CF
  - IOFFDAC_LED

Driver Framework:

Input parameters for low level functions are pre defined and matches datasheet

- 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 };
- enum **REG_TWLED** {LED_ON_16uS = 1, LED_ON_24uS = 2, LED_ON_31uS = 3, LED_ON_39uS = 4, LED_ON_47uS = 5, LED_ON_63uS = 7, LED_ON_70uS = 8, LED_ON_78uS = 9, LED_ON_94uS = 11, LED_ON_117uS = 14};

Driver Framework:

- **Since all the functions are written in embedded C, it will be MCU independent.**
- **Everything breaks down to two functions (i.e. writeReg and readReg)**

```
void AFE_config_phaseTimingScheme(phTmgScheme)
{
   if( phTmgScheme == HIGH_PRF_MODE)  {
      AFE_modifyRegGlobal(&dev1.GLOBAL.HIGH_PRF_MODE,                true);
      AFE_modifyRegGlobal(&dev1.GLOBAL.MAX_AMB_REJ,                  false);
      AFE_modifyRegGlobal(&dev1.GLOBAL.DIS_POST_AMB_MAX_AMB_REJ,  false);
   }
   else if (phTmgScheme == MAX_AMB_REJ)  {
      AFE_modifyRegGlobal(&dev1.GLOBAL.HIGH_PRF_MODE,                false);
      AFE_modifyRegGlobal(&dev1.GLOBAL.MAX_AMB_REJ,                  true);
      AFE_modifyRegGlobal(&dev1.GLOBAL.DIS_POST_AMB_MAX_AMB_REJ,  false);
   }
            ........

}
```

AFE_Parameter has 3 attributes

- **Address**
- **MSB Bit**
- **LSB Bit**

} Forms the MaskPattern

```
void AFE_modifyRegGlobal (AFE_Parameter, Value)
{
    readValue = readReg(AFE_Parameter.Address);        // For consecutive register writes of same address, read happens only once.
    writeValue = readValue & MaskPattern + Value;
    writeReg(AFE_Parameter.Address, writeValue);       // Writes only when write vale is not same as read value.
}
```

Register write/read function

```
void writeReg (Address, Data)
{
        I2C/SPI_Write(Address, Data);
}
```

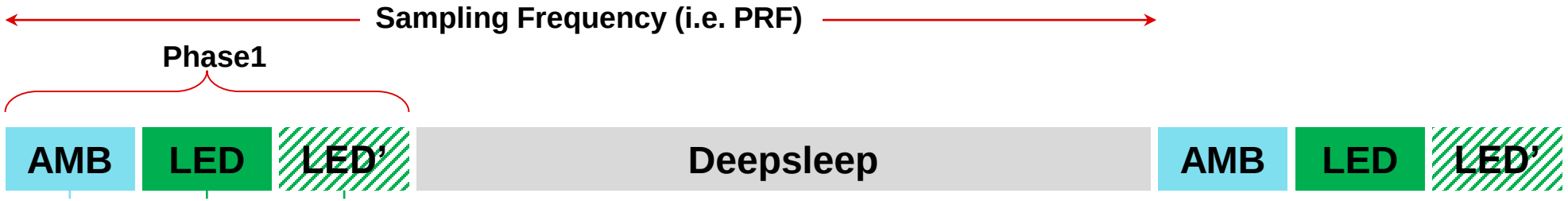
```
Data readReg (Address)
{
        readData = I2C/SPI_Read(Address);
        return (readData);
}
```

- **I2C/SPI_Write and I2C/SPI_Read are MCU dependent.**
- **Customer needs to update these two functions with appropriate function call based on their MCU.**

Example1: HRM Configuration

Typical system requirements:

- Green LED is used along with IR-cut PD (IR-cut to suppress IR wavelength in ambient such as sunlight). ***1 LED Phase***
- Remove the DC Steady part and amplify AC Pulsatile part of the PPG signal. ***LED DC Cancellation***
- Estimate and cancel any ambient signal from PPG signal. ***ANA_AACM***
- Low power consumption for continuous monitoring.



On-demand LED sub-phase inserted by the AFE when output of LED sub-phase exceeds the programmed threshold as part of automatic LED DC cancellation with DRE.

Phase being defined as LED.

Auto inserted pre ambient sub-phase to estimate and cancel AMB. FIFO is stored with LED – AMB to further suppress AMB in PPG signal.

HRM Configuration Example

System Requirements:

Internal OSC mode.
PRF of 25Hz.
LED ON time of 117uS.
RF = 250KOhm.

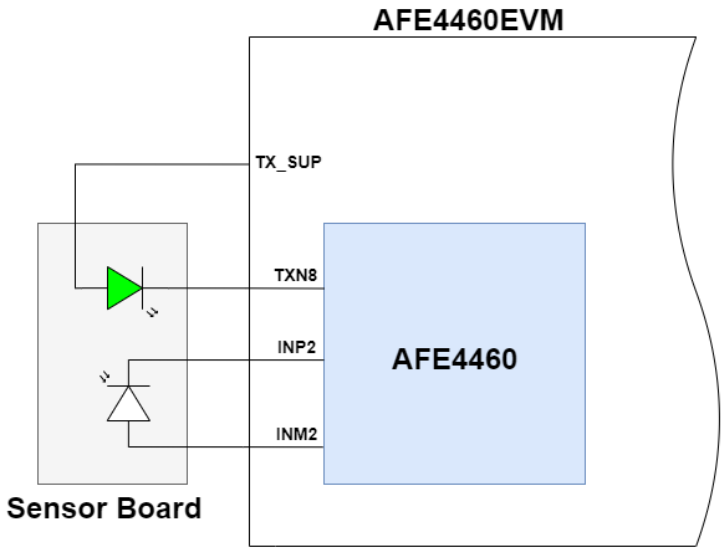
Green LED connected to TXN8.

LED-AMB data streams into FIFO.
Averaging of 2 ADC samples.

PPG is sampled from PD2 (i.e. INP2 and INM2) and LED's DC is automatically cancelled.

AMB is cancelled using ANA_AACM.

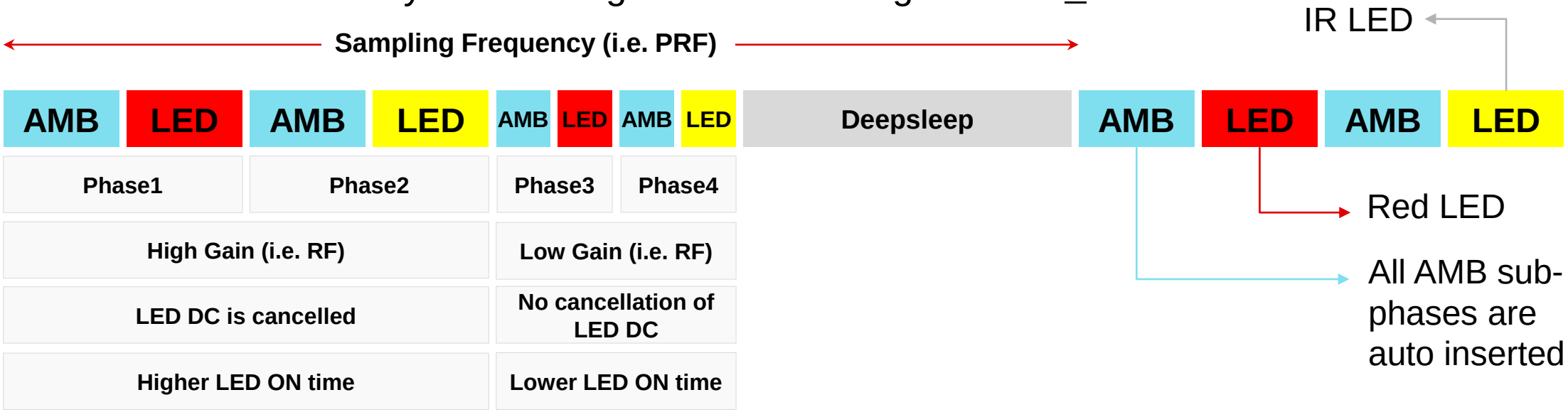
```
void AFE_configAFE_forHRM(){

    uint8_t PhaseToConfig;
    AFE_config_regMapInit();
    AFE_set_SW_RESET();
    AFE_clearPPM();
    AFE_initializeAFE();
    AFE_config_phaseTimingScheme(STAGGER);
    AFE_config_clockMode(CLK_MODE_INT);
    AFE_set_PRPCT(10240);
    AFE_set_FILTER_BW(LED_ON_117uS, LED_ON_117uS);
    AFE_set_REG_SCALE_DRE(RF_250KOhm);

    AFE_modifyRegGlobal(&dev1.GLOBAL.REG_NUMPHASE, 0);
    PhaseToConfig = Phase1;
    AFE_set_phaseType( PhaseToConfig, LED_WithPreAMB, TXN8, TXN8, TX_SUP);
    AFE_set_REG_TWLED( PhaseToConfig, LED_ON_117uS);
    AFE_set_ILED_DRVx( PhaseToConfig, 50, 50);
    AFE_set_FIFO_DATA_CTRL( PhaseToConfig, LED_AMB);
    AFE_set_NUMAV( PhaseToConfig, 2);
    AFE_configTIA( PhaseToConfig, TIA1, PD2, RF_250KOhm,LED_ON_117uS, LED_cancelWithDRE);

    AFE_config_AMBCancellation(PhaseToConfig, AMB_estimateAndCancel);
    AFE_modifyRegGlobal(&dev1.GLOBAL.REG_WM_FIFO, 9);
    if (errorFlags==0)
        AFE_enableTimingEngine();
}
```



Example2: SpO2 Configuration

Typical system requirements:

- Red and IR LEDs used along with broadband PD.
- Simultaneously measure DC Steady part and AC Pulsatile part of the PPG signal. 2 phases for each wavelength of LED.
  - High Gain Phase: Amplifies the Pulsatile part of PPG after cancelling Steady part using LED DC Cancellation
  - Low Gain Phase: Gain is lowered to support Steady Part of PPG.
- Estimate and cancel any ambient signal from PPG signal. ***ANA_AACM***



SpO2 Configuration Example

System Requirements:

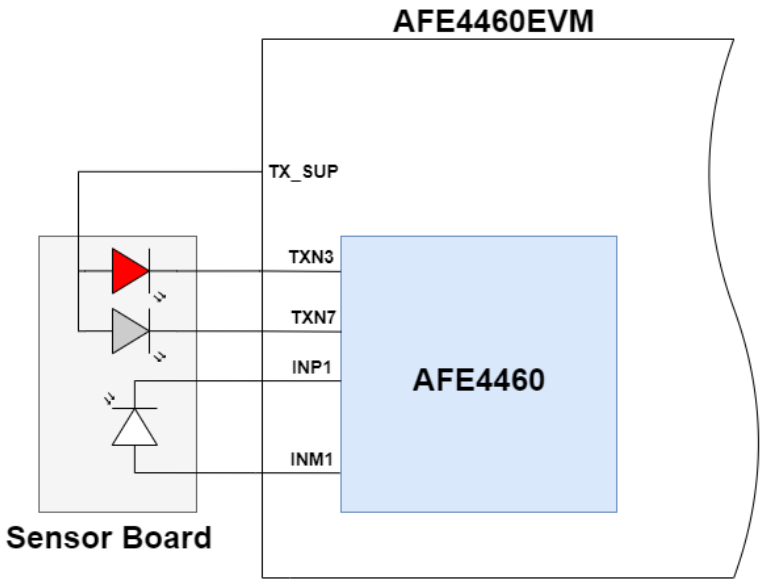
Internal OSC mode.
PRF of 100Hz.
2 FILT BWs due to 2 LED ON times

Red LED to acquire AC Signal

Red LED connected to TXN3.
LED ON time of 117uS.

Red LED sampled on PD1,RF = 250KOhm, auto cancellation of DC
AMB is estimated and cancelled

```
void AFE_configAFE_forSpO2(){

    uint8_t PhaseToConfig;
    enum REG_TWLED LED_ON_AC = LED_ON_117uS;
    enum REG_TWLED LED_ON_DC = LED_ON_16uS;
    enum RF_TIA rfSelected_AC = RF_250KOhm;
    enum RF_TIA rfSelected_DC = RF_10KOhm;
    AFE_config_regMapInit();
    AFE_set_SW_RESET();
    AFE_clearPPM();
    AFE_initializeAFE();
    AFE_config_phaseTimingScheme(HIGH_PRF_MODE);
    AFE_config_clockMode(CLK_MODE_INT);
    AFE_set_PRPCT(2560);
    AFE_set_FILTER_BW(LED_ON_AC, LED_ON_DC);
    AFE_set_REG_SCALE_DRE(rfSelected_AC);

    AFE_modifyRegGlobal(&dev1.GLOBAL.REG_NUMPHASE, 3);

    PhaseToConfig = Phase1;          // Red LED for AC component of PPG
    AFE_set_phaseType(PhaseToConfig, LED_WithPreAMB, TXN3, TXN3, TX_SUP);
    AFE_set_REG_TWLED( PhaseToConfig, LED_ON_AC);
    AFE_set_ILED_DRVx( PhaseToConfig, 50, 50);
    AFE_set_FIFO_DATA_CTRL(PhaseToConfig, LED_AMB);
    AFE_set_NUMAV( PhaseToConfig, 2);
    AFE_configTIA( PhaseToConfig, TIA1, PD1, rfSelected_AC,LED_ON_AC, LED_cancelWithoutDRE);
    AFE_config_AMBCancellation( PhaseToConfig, AMB_estimateAndCancel);
```



IR LED to acquire AC Signal

IR LED connected to TXN7.
LED ON time of 117uS.

IR LED sampled on PD1,RF = 250KOhm, auto cancellation of DC
AMB is cancelled

Red LED to acquire DC Signal

Red LED connected to TXN3.
LED ON time of 16uS.

Red LED sampled on PD1,RF = 10KOhm, no cancellation of DC
AMB is estimated and cancelled

IR LED to acquire DC Signal

Red LED connected to TXN3.
LED ON time of 16uS.

IR LED sampled on PD1,RF = 10KOhm, no cancellation of DC
AMB is cancelled

```
    PhaseToConfig = Phase2;          // IR LED for AC component of PPG
    AFE_set_phaseType(PhaseToConfig, LED_WithPreAMB, TXN7, TXN7, TX_SUP);
    AFE_set_REG_TWLED(PhaseToConfig, LED_ON_AC);
    AFE_set_ILED_DRVx(PhaseToConfig, 50, 50);
    AFE_set_FIFO_DATA_CTRL(PhaseToConfig, LED_AMB);
    AFE_set_NUMAV(PhaseToConfig, 2);
    AFE_configTIA(PhaseToConfig, TIA1, PD1, rfSelected_AC,LED_ON_AC, LED_cancelWithoutDRE);
    AFE_config_AMBCancellation(PhaseToConfig, AMB_cancel);

    PhaseToConfig = Phase3;          // Red LED for DC component of PPG
    AFE_set_phaseType(PhaseToConfig, LED_WithPreAMB, TXN3, TXN3, TX_SUP);
    AFE_set_REG_TWLED(PhaseToConfig, LED_ON_DC);
    AFE_set_ILED_DRVx(PhaseToConfig, 50, 50);
    AFE_set_FIFO_DATA_CTRL(PhaseToConfig, LED_AMB);
    AFE_set_NUMAV(PhaseToConfig, 2);
    AFE_configTIA(PhaseToConfig, TIA1, PD1, rfSelected_DC,LED_ON_DC, LED_cancelDisabled);
    AFE_config_AMBCancellation(PhaseToConfig, AMB_estimateAndCancel);

    PhaseToConfig = Phase4;          // IR LED for DC component of PPG
    AFE_set_phaseType(PhaseToConfig, LED_WithPreAMB, TXN7, TXN7, TX_SUP);
    AFE_set_REG_TWLED(PhaseToConfig, LED_ON_DC);
    AFE_set_ILED_DRVx(PhaseToConfig, 50, 50);
    AFE_set_FIFO_DATA_CTRL(PhaseToConfig, LED_AMB);
    AFE_set_NUMAV(PhaseToConfig, 2);
    AFE_configTIA(PhaseToConfig, TIA1, PD1, rfSelected_DC,LED_ON_DC, LED_cancelDisabled);
    AFE_config_AMBCancellation(PhaseToConfig, AMB_cancel);
    AFE_modifyRegGlobal(&dev1.GLOBAL.REG_WM_FIFO, 39);
    if (errorFlags==0)
        AFE_enableTimingEngine();
}
```

Guidelines:

- Certain values of LED ON times are supported so that
  - CF can be automatically calculated based on RF and LED ON time. // **“AFE_compute_CF”**
  - Noise Reduction Filters are automatically configured based on LED ON time. // **“AFE_set_FILTER_BW”**

Following table is used as reference for CF and bandwidth calculations.

LED ON time, us	FILTER_BW_PRE, kHz	FILTER_BW_FINE, kHz	tW_FILTER_PRE, tTE	Value of CF used, pF													
				RF = 3.7 KΩ	RF = 5 KΩ	RF = 10 KΩ	RF = 25 KΩ	RF = 33.3 KΩ	RF = 50 KΩ	RF = 71.5 KΩ	RF = 100 KΩ	RF = 142 KΩ	RF = 166 KΩ	RF = 200 KΩ	RF = 250 KΩ	RF = 500 KΩ	RF = 1000 KΩ
16	50	50	0	25	25	25	25	25	25	25	25	20	17.5	10	10	5	2.5
24	35	35	0	25	25	25	25	25	25	25	25	25	25	22.5	17.5	7.5	2.5
31	25	25	0	25	25	25	25	25	25	25	25	25	25	25	22.5	10	5
39	50	15	4	25	25	25	25	25	25	25	25	20	17.5	10	10	5	2.5
47	50	10	4	25	25	25	25	25	25	25	25	20	17.5	10	10	5	2.5
63	50	7.5	4	25	25	25	25	25	25	25	25	20	17.5	10	10	5	2.5
70	50	5	4	25	25	25	25	25	25	25	25	20	17.5	10	10	5	2.5
78	32.5	5	6	25	25	25	25	25	25	25	25	25	25	25	20	10	5
94	32.5	5	6	25	25	25	25	25	25	25	25	25	25	25	20	10	5
102	32.5	5	6	25	25	25	25	25	25	25	25	25	25	25	20	10	5
117	25	2.5	8	25	25	25	25	25	25	25	25	25	25	25	22.5	10	5

If different LED ON time is needed then user has to compute the CF and bandwidths based on the datasheet.

- REG_SCALE_DRE is automatically set based on RF. User needs to pass the largest value of RF used (with the phases where DRE is enabled) to the **“AFE_set_REG_SCALE_DRE”** function.

	RF = 1000 KΩ	RF = 500 KΩ	RF = 250 KΩ	RF = 200 KΩ	RF = 166 KΩ	RF = 142 KΩ	RF = 100 KΩ	RF = 71.5 KΩ	RF = 50 KΩ	RF = 33.3 KΩ	RF = 25 KΩ	RF = 10 KΩ
Extension factor for DRE	32	32	16	32	32	32	16	16	8	8	4	4

Guidelines:

- There are few checks within the driver and its outcome can be read out using variable called **“errorFlags”**. Each bit of **“errorFlags”** denotes a certain error as described below.

Bit location of errorFlags	Error Description	Fix/Comment
D0	Different LED ON time is used instead of pre-defined ones	Compute CF and bandwidth for noise reduction filters manually based on datasheet.
D1	Error due to AUTO_AMB_INSERT	Choose AUTO_AMB_INSERT based on Phase Timing Scheme
D2	DRE being enabled in HIGH_PRF_MODE	DRE is not supported in HIGH_PRF_MODE
D3	Error due to FIFO_DATA_CTRL	Choose FIFO_DATA_CTRL based on Phase Timing Scheme and ENABLE_DRE

“errorFlags” gets reset to 0 after software reset. Bits do not reset to 0 after the error is fixed so its advised to call software reset at the beginning of configuration and check **“errorFlags”** when all the parameters are configured to make sure there are no errors.

How to write new configurations:

1. Call 4 mandatory functions

2. Set Phase Timing Scheme

3. Set Clock Mode

4. Set PRF as PRPCT = fCLK_PRF / PRF

5. Define LED ON times and configure FILTER_BW

6. Define Number of Phases required.

   Global (i.e. Page0) registers can be accessed using “AFE_modifyRegGlobal”

7. Configure each Phase.

   Per-Phase (i.e. Page1) registers can be accessed using “AFE_modifyRegPPM”

   Set LED – PD association, LED ON time, RF, LED current, NUMAV, FIFO_DATA_CTRL, schemes for ambient and LED’s DC cancellation

8. Set Water Mark level for FIFO

9. Check for “errorFlags” and enable the timing engine

Verify all the configurations in this function

```
// Initialization and default function calls.
AFE_config_regMapInit();                                  // Mandatory Step1
AFE_set_SW_RESET();                                       // Mandatory Step2
AFE_clearPPM();                                           // Mandatory Step3
AFE_initializeAFE();                                      // Mandatory Step4,
AFE_config_phaseTimingScheme(STAGGER);                    // Phase timing sch
AFE_config_clockMode(CLK_MODE_INT);                       // CLK_MODE_INT is
AFE_set_PRPCT(10240);                                     // PRPCT of 10240 c
AFE_set_FILTER_BW(LED_ON_117uS, LED_ON_117uS);            // Both set of Nois
AFE_set_REG_SCALE_DRE(RF_250KOhm);                        // Max value of Rf

AFE_modifyRegGlobal(&dev1.GLOBAL.REG_NUMPHASE, 0);  // Only 1 phase to

PhaseToConfig = Phase1;
AFE_set_phaseType(              PhaseToConfig, LED_WithPreAMB, TXN8, TX
AFE_set_REG_TWLED(              PhaseToConfig, LED_ON_117uS);
AFE_set_ILED_DRVx(              PhaseToConfig, 50, 50);
AFE_set_FIFO_DATA_CTRL(         PhaseToConfig, LED_AMB);
AFE_set_NUMAV(                  PhaseToConfig, 2);
AFE_configTIA(                  PhaseToConfig, TIA1, PD2, RF_250KOhm,LE
AFE_config_AMBCancellation(     PhaseToConfig, AMB_estimateAndCancel);

AFE_modifyRegGlobal(&dev1.GLOBAL.REG_WM_FIFO, 9);

if (errorFlags==0)
    AFE_enableTimingEngine();
```