Detailed Design (DD)

Timer0_OverFlow Mode

Authors: Burners Team (Ahmed Salah, Basem Moufreh, Hassan El Gabass , Hazem El Morshedi , Mohamed Safwat).

Customer: NTI

Instructor: Mahmoud Ali, Ahmed Abd El Reheem.

APIs

1. Initialization API
2. Set Callback API
3. Set Preload API

1 Initializing API

Name	Initialization Function
Prototype	STD MCAL_Timer0_init(void);
Parameter in	-
Parameter out	-
Parameter in-out	-
Return Type	STD E_OK \ STD E_NOT_OK
Description	This Function is responsible of initializing the Timer0: * 1. Setup the timer0 in overflow mode (normal mode) * 2. Set the appropriate prescaler * 3. Enable the timer
Covered Requirement	[SRS_TIMER0_2.1], [SRS_TIMER0_2.2], [SRS_TIMER0_3.1], [SRS_TIMER0_3.2], [SRS_TIMER0_3.3].

2 Set callback API

Name	Set callback function
Prototype	void TIMER0_SetCallBack_OVF (void (*Local_PointerToFunction_OVF)(void))
Parameter in	void (*Local_PointerToFunction_OVF) (void)
Parameter out	-
Parameter in-out	-
Return Type	STD E_OK \ STD E_NOT_OK
Description	This Function sets the callback function to execute when the overflow happens.
Covered Requirement	[SRS_TIMER0_2.3], [SRS_TIMER0_2.4]. [SRS_TIMER0_2.5], [SRS_TIMER0_3.1],[SRS_TIMER0_3.2], [SRS_TIMER0_3.3].

3 Set preload API

Name	Set preload value
Prototype	void TIMER0_u8SetPreloadValue(u8 preloadValue);
Parameter in	u8 preloadValue
Parameter out	-
Parameter in-out	-
Return Type	STD E_OK \ STD E_NOT_OK
Description	This Function is responsible of: set the preload value
Covered Requirement	[SRS_TIMER0_2.4], [SRS_TIMER0_3.1], [SRS_TIMER0_3.2], [SRS_TIMER0_3.3].