/\*

\* TIMER\_prg.c

\*

\* Created on: Jul 27, 2025

\* Author: Administrator

\*/

#include "../../LIB/BIT\_MATH.h"

#include "TIMER\_int.h"

#include "TIMER\_prv.h"

#include "TIMER\_cfg.h"

u32 G\_u32Counter = 0;

void(\*G\_fsetCallBack) (void)= ((void\*)(0));

void MTIMER\_vInit(void){

// choose your mode

TCCR0 = 0b00000000;

// enable interrupt

SET\_BIT(TIMSK,TOIE0);

}

void MTIMER\_vSetPreload(u8 A\_u8Preload){

TCNT0 = A\_u8Preload;

}

void MTIMER\_vStart(void) {

TCCR0 = (TCCR0 & 0b11111000) | (PRESCALER\_TIMER & 0b00000111);

}

void MTIMER\_vStop(void) {

TCCR0 = (TCCR0 & 0b11111000) | (0b000);

}

void MTIMER\_vSetInternalVal(u32 A\_u32NumOfTick) {

G\_u32Counter = A\_u32NumOfTick;

}

void MTIMER\_vSetCallBackAddress(void (\*A\_fptr) (void)) {

G\_fsetCallBack = A\_fptr;

}

// no need to clear flag again it does so automatically when entering ISR -> mensioned in data sheet

void \_\_vector\_11(void) \_\_attribute\_\_((signal));

void \_\_vector\_11(void)

{

static u32 count = 0; // to be made global or static local

count++;

// when count reaches my desire he do this

if(count >= G\_u32Counter ) {

count = 0;

G\_fsetCallBack();

}

}