Task

AUTHOR
Version
26/10/2020

Table of Contents

Table of contents

Module Index

Modules

Here is a list of all modules:

File Index

File List

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

Module Documentation

main.c

Timer_example Application main file.

Macros

#define **TIMEOUT_US** 509 * 1000

Functions

void **increment_count_event_handler** (nrf_timer_event_t event_type, void *p_context)

Handler for timer events.

void **timer_stop** ()
void **timer_start** (uint32_t time_us, void *event_handler)
int **main** (void)

Function for application main entry. it prints the count after the signal from event handler is set.

Variables

const nrf_drv_timer_t **TIMER_COUNTER** = NRF_DRV_TIMER_INSTANCE(0)
uint8_t **signal** =0

Detailed Description

Timer_example Application main file.

This file contains the source code for a simple application that prints a counter value that increments every 509 ms

---

Function Documentation

void increment_count_event_handler (nrf_timer_event_t *event_type*, void * *p_context*)

Handler for timer events.

Parameters

in	*event_type*	Channel 0 is set so, verifying that event only
in	*p_context*	It has the value till that timer should run

Returns

none

File Documentation

main.c File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include "nrf.h"
#include "nrf_drv_timer.h"
#include "nrf_delay.h"
#include "nrf_log.h"
#include "boards.h"
```

Include dependency graph for main.c:

IMAGE

Macros

#define **TIMEOUT_US** 509 * 1000

Functions

void **increment_count_event_handler** (nrf_timer_event_t event_type, void *p_context)

Handler for timer events.

void **timer_stop** ()
void **timer_start** (uint32_t time_us, void *event_handler)
int **main** (void)

Function for application main entry. it prints the count after the signal from event handler is set.

Variables

const nrf_drv_timer_t **TIMER_COUNTER** = NRF_DRV_TIMER_INSTANCE(0)
uint8_t **signal** =0
