

Universidade Federal de Santa Maria

Centro de Tecnologia

DISPOSITIVO PARA ENCONTRAR SINAIS DE BLUETOOTH
DOCUMENTAÇÃO DO CÓDIGO FONTE

Felipe Seitenfus, Leandro Frazzon e William Azevedo

Disciplina: Projeto de Sistemas Embarcados

Quarta-feira, 28 de Junho de 2017

Documentação do ficheiro

Funções

void **tc_cc0_cb** (struct tc_module *const module_inst)

Função de configuração do timer.

void **configure_eeprom** (void)

Configuração da memória EEPROM.

static void **configure_bod** (void)

Configuração da interrupção da memória EEPROM, juntamente com a função SYSCTRL_Handler(void).

static void **timer_callback_handler** (void)

Tratamento da interrupção do timer.

static void **app_immediate_alert** (uint8_t alert_val)

Tratamento do sinal Bluetooth mandado pelo dispositivo externo ao periférico para o dispositivo.

PT_THREAD (pt_find_me(struct pt *pt, char data))

Protothread.

int **main** (void)

Função main.

Variáveis

static struct usart_module **cdc_uart_module**

struct que guarda a configuração da porta serial.

struct tc_config **config_tc**

struct que guarda a configuração do timer.

struct usart_config **usart_conf**

struct que guarda a configuração da porta serial.

static const ble_event_callback_t **fmp_gap_handle** []

interrupções aceitas pelo dispositivo na aplicação

static const ble_event_callback_t **fmp_gatt_server_handle** []

interrupções aceitas pelo dispositivo na aplicação

uint32_t **timeout_count**

recebe o sinal alto ou médio do LED.

hw_timer_callback_t **timer_callback**

recebe a interrupção do timer.

at_ble_events_t **event**

variável que trata dos eventos do dispositivo Bluetooth.

uint8_t **ble_event_params** [524]

variável que trata dos eventos do dispositivo Bluetooth.

uint8_t **last_alert** = 0

variável que recebe o último alerta dado por um dispositivo externo ao periférico.

uint8_t **page_data** [EEPROM_PAGE_SIZE]

variável de comunicação entre a memória e o programa.

volatile char **i** = 0

variável de comunicação entre a protothread e a função Main

volatile char **buffer**

variável de comunicação entre a protothread e a função Main

gatt_service_handler_t **ias_handle**

variável que especifica o uso do dispositivo Bluetooth para a função de rastreamento

volatile bool **app_timer_done** = false

Flag da tarefa do Timer.

static uint8_t **timer_interval** = INIT_TIMER_INTERVAL

Flag da contagem de tempo.

find_me_callback_t **immediate_alert_cb**

Interrupção do aviso imediato.

static struct pt **pt**

struct que trata a protothread.

Descrição detalhada

Programa principal para encontrar dispositivos Bluetooth.

Utilizaremos a placa SAMD21, que possui compatibilidade com essa forma de comunicação de dados. A captação dos sinais Bluetooth é feita através do periférico ATBTLC1000, que é do tipo BLE (*Bluetooth Low Energy*). Após encontrar um dispositivo Bluetooth nas proximidades da placa e do periférico, o nome registrado será armazenado na memória EEPROM da própria SAMD21.

Veja também:

asf.h

usart.h

platform.h

timer_hw.h

tc_interrupt.h

conf_timer.h

conf_extint.h

ble_manager.h

immediate_alert.h

find_me_app.h

find_me_target.h

pt.h

Documentação das funções

void app_immediate_alert (uint8_t alert_val)[static]

Tratamento do sinal Bluetooth mandado pelo dispositivo externo ao periférico para o dispositivo.

Ele indica se o sinal mandado foi alto (High), médio (Mild) ou fraco (No). Se o sinal for High ou Mild, o LED acende. Se o sinal for No, apaga. Em seguida, é feita a gravação do último sinal dado durante a execução do aplicativo na memória.

Parâmetros:

<i>alert_val</i>	sinal mandado pelo dispositivo externo ao periférico.
------------------	---

void configure_bod (void) [static]

Configuração da interrupção da memória EEPROM, juntamente com a função SYSCTRL_Handler(void).

void configure_eeprom (void)

Configuração da memória EEPROM.

EEPROM é uma emulação de uma memória que depende da configuração dos fusos. O tamanho dela na placa deve ser menor que cinco. Se a configuração dos fusos da memória não foi feita da maneira correta. Então, o programa fica em um loop infinito. Se a EEPROM estiver mal configurada ou corrompida, ela é resetada e seus dados são apagados.

int main (void)

Função *main*.

Inicializando a protothread com PT_INIT().

PT_THREAD (pt_find_me(struct pt *pt, char data))

Protothread.

A protothread *pt_find_me* é responsável por configurar e executar a aplicação. Além da SAMD21, o controle serial, o timer, o dispositivo Bluetooth e as interrupções são inicializadas e configuradas aqui. Ela não retorna para a função main, já que o modo de espera também é executado pela protothread.

A função *system_init()* inicializa o sistema.

A Inicialização do controle serial é feita pela função *stdio_serial_init(&cdc_uart_module, SERCOM3, &usart_conf)*.

O timer é iniciado pela função *tc_init(&tc_instance, TC3, &config_tc)*. Um counter size de 32 bits permite um tempo maior de espera, dado que o sinal Bluetooth é instável.

As funções *configure_eeprom()* e *configure_bod()* configuram a memória EEPROM.

A função *ble_device_init(NULL)* inicia o dispositivo Bluetooth.

A Inicialização do serviço de rastreamento é feita por *init_immediate_alert_service(&ias_handle)*.

A função *init_immediate_alert_service(&ias_handle)* é responsável pela definição da aplicação para o dispositivo Bluetooth. Indica que ele será usado para o serviço de rastreamento.

A função *at_ble_adv_start()* dá o status do dispositivo. Advertising Mode é o modo de espera do dispositivo após a inicialização. Até que um celular seja conectado com sucesso, ele fica nesse modo.

A função *ble_mgr_events_callback_handler(REGISTER_CALL_BACK, BLE_GAP_EVENT_TYPE, fmp_gap_handle)* faz o Registro de interrupção do BLE-GAP.

O Registro de interrupção do BLE-GATT-Server é feito por *ble_mgr_events_callback_handler(REGISTER_CALL_BACK,*

ble_event_manager(event, ble_event_params), ele fica no aguardo de uma interrupção. Quando ela for dada, a função *app_immediate_alert* é chamada. Caso o limite de tempo seja atingido sem que um dispositivo tenha mandado um sinal, ele volta para o Advertising Mode.

Parâmetros:

<i>pt_find_me(s</i> <i>truct</i>	pt *pt, char data) função responsável pela execução na protothread.
-------------------------------------	---

void tc_cc0_cb (struct tc_module *const module_inst)

Função de configuração do timer.

void timer_callback_handler (void)[static]

Tratamento da interrupção do timer.

O timer é desabilitado. Se *app_timer_done* é true, é uma indicação de que nenhum dispositivo foi conectado no tempo possível, ou não conseguiu conectar. Então, ele reinicia no modo Advertising Mode.

Documentação das variáveis

volatile bool app_timer_done = false

Flag da tarefa do Timer.

uint8_t ble_event_params[524]

variável que trata dos eventos do dispositivo Bluetooth.

volatile char buffer

variável de comunicação entre a protothread e a função Main

```
struct usart_module cdc_uart_module[static]
```

struct que guarda a configuração da porta serial.

struct tc_config config_tc

struct que guarda a configuração do timer.

at ble events t event

variável que trata dos eventos do dispositivo Bluetooth.

```
const ble_event_callback_t fmp_gap_handle[][static]
```

[illegible]

interrupções aceitas pelo dispositivo na aplicação

```
const ble_event_callback_t fmp_gatt_server_handle[][static]
```

```
Valor inicial:= {  
    NULL,  
    NULL,  
    fmp_target_char_changed_handler,  
    NULL,  
    NULL,  
    NULL,
```

```
NULL,  
NULL,  
NULL,  
NULL  
}
```

interrupções aceitas pelo dispositivo na aplicação

volatile char i = 0

variável de comunicação entre a protothread e a função Main.

gatt_service_handler_t ias_handle

variável que especifica o uso do dispositivo Bluetooth para a função de rastreamento.

find_me_callback_t immediate_alert_cb

Interrupção do aviso imediato.

uint8_t last_alert = 0

variável que recebe o último alerta dado por um dispositivo externo ao periférico.

uint8_t page_data[EEPROM_PAGE_SIZE]

variável de comunicação entre a memória e o programa.

struct pt pt[static]

struct que trata a protothread.

uint32_t timeout_count

recebe o sinal alto ou médio do LED.

hw_timer_callback_t timer_callback

recebe a interrupção do timer.

uint8_t timer_interval = INIT_TIMER_INTERVAL[static]

Flag da contagem de tempo.

struct usart_config usart_conf

struct que guarda a configuração da porta serial.