

CoAP- Server/Aktuator -Labor

Distributed Systems, SS24, Sigmund/Dornfeld, Prof. Böck

Ziel des Versuches:

In diesem Laborversuch soll ein CoAP-Aktor angesteuert werden.

Im Weiteren werden verschiedene Codefragmente vorgegeben, allerdings ist nur eines davon richtig (Multiple Choice). Das richtige Codefragment muss dann an der richtigen Stelle im C- Code eingefügt werden.

Die Aktoren die angesteuert werden ist zum einen der Buzzer und zum anderen die PWM LED auf dem Boosterpack Board.

Damit es funktioniert muss das Boosterpack Board auf den Steckplatz 1 gesteckt werden.

Im Ordner ist ein CoAP- Client der in Python geschrieben ist. Wird dieser ausgeführt sollte ein „Klavier“ erscheinen. Mit diesem kann der Buzzer auf dem Boosterpack Board angesteuert werden.

Um das Python Programm ausführen zu können werden folgende zusätzliche Bibliotheken benötigt die gegebenenfalls installiert werden müssen.

- tkinter
- aicoap

Befehl: *pip3 install <Bibliotheksname>*

Später muss dann noch die richtige IP- Adresse des CoAP- Servers im Python Client angepasst werden.

CCS- Programm

Im Projekt müssen nach dem importieren die Variablen „SW_ROOT“ und „SW_FREERTOS“ in den Properties angepasst werden.

SW_ROOT: *TivaWare_C_Series-2.2.0.295*

SW_FREERTOS: *TivaWare_C_Series-2.2.0.295/third_party*

Multiple Choice 1:

An dieser Stelle sollte ein sehr wichtiger Include eingefügt werden. Diese ist für die Implementierung des CoAP- Protokolls essentiell.

- a) `#include "libCoAP.h"`
- b) `#include "mongoose.h"`
- c) `#include "coap.h"`

Multiple Choice 2:

An dieser Stelle muss der richtige TCP/IP Stack eingefügt werden. Dieser ist unter anderem für das DHCP verantwortlich, welches wir benutzen wollen.

- a) `#include "utils/lwiplib.h"`
- b) `#include "utils/UDPIplib.h"`
- c) `#include "utils/TCPIplib.h"`

Multiple Choice 3:

Hier muss die **Standard** Adresse von unserem CoAP- Port angegeben werden.

- a) `static char *s_default_address = "udp://:5683";`
- b) `static char *s_default_address = "udp://:5684";`
- c) `static char *s_default_address = "udp://:8883";`

Multiple Choice 4:

Schaue dir die Logik der Funktion *lwIPHostTimerHandler* genau an und entscheide welche Zeile sinnvoll ist.

- a) `uint32_t ui32NewIPAddress = lwIPLocalIPAddrSend();` // Aktuelle IP-Adresse wird verschickt
- b) `uint32_t ui32NewIPAddress = lwIPLocalMacAddrGet();` // Aktuelle MAC Adresse wird abgerufen und zugewiesen
- c) `uint32_t ui32NewIPAddress = lwIPLocalIPAddrGet();` // Aktuelle IP-Adresse wird abgerufen und zugewiesen

Multiple Choice 5:

Nun initialisieren wir den CoAP-Handler. Die Funktion benötigt eine 3 der 5 Übergabeparameter. Die Übergabeparameter werden in der folgenden Reihenfolge benötigt:

```
static void coap_handler(mongoose Verbindung, Event, Event Daten){
```

Setze die 3 richtigen Parameter ein.

- a) struct mg_connection *nc
- b) int mg
- c) void *ev_data
- d) void data
- e) int ev

Tipp: Schaue die ersten Zeilen des CoAP-Handlers an.

Multiple Choice 6:

Als nächstes wird der passende case benötigt.

- a) COAP_CODE_GET:
- b) COAP_CODE_POST:
- c) COAP_CODE_PUT:
- d) COAP_CODE_DELETE:

Multiple Choice 7:

Nun muss die lwIP- Bibliothek initialisiert werden.

- a) lwIPInit(g_ui32SysClock, pui8MACArray, 0, 0, 0, IPADDR_USE_AUTOIP);
- b) lwIPInit(g_ui32SysClock, pui8MACArray, 0, 0, 0, IPADDR_USE_DHCP);
- c) lwIPInit(g_ui32SysClock, pui8MACArray, 0, 0, 0, IPADDR_USE_STATIC);

Tipp: Wenn du nicht drauf kommst schau dir nochmal eine der ersten Multiple Choice Fragen an.

Multiple Choice 8:

Nun wird der Mongoose Manager initialisiert. Hier muss die passende Struktur übergeben werden.

- a) mg_mgr_init(&g_mgr, NULL);
- b) mg_mgr_init(&nc,NULL);
- c) mg_mgr_init(&g_gr,NULL);

Multiple Choice 9:

Nun binden wir unseren Mongoose Manager, unseren CoAP-Handler an die vorher definierte Adresse und an die Verbindungsstruktur von Mongoose übergeben.

- a) `ab = mg_bind(&g_mgr, s_default_address, coap_handler);`
- b) `co = mg_bind(&g_mgr, s_default_address, coap_handler);`
- c) `nc = mg_bind(&g_mgr, s_default_address, coap_handler);`

Multiple Choice 10:

Jetzt muss das richtige Protokoll eingestellt werden.

- a) `mg_set_protocol_mqtt(nc);`
- b) `mg_set_protocol_coap(nc);`
- c) `mg_set_protocol_xmpp(nc);`

Multiple Choice 11:

Nun muss der Mongoose Manager in regelmäßigen Abständen auf eingehende CoAP-Nachrichten überprüft werden.

- a) `mg_mgr_poll(&du, 0);`
- b) `mg_mgr_poll(&hast, 0);`
- c) `mg_mgr_poll(&es, 0);`
- d) `mg_mgr_poll(&geschafft, 0);`
- e) `mg_mgr_poll(&g_mgr, 0);`

Viel Spaß mit deinem Klavier!!!

PS: Du kannst auch mit den Tasten A-K spielen.

Versuch ´s doch mal mit „Alle meine Entchen“

