

C Project Example

This project have been made by using the STM32Cube library. It is a middleware that provided the interfaces to the components presents on the boards. I.e. setting up specific bits into the registries.

In the src folder of each project can be found all the code i've made. To be noted, the files:

- _initialize_hardware.c
- _write.c
- stm32f4_discovery.c
- stm32f4xx_hal_msp.c

are provided by the library and can be edited to have some kind of flexibility during the configuration on specific boards.

STM32 Bus Exercises

In this folder there's different kind of exercises where the focus is on the use of the middleware to communicate.

The exercises have different names depending on the configurations:

- Master & slave -> communication with 2 different boards
- OnBoard -> communication with a single board using different components of the same type
- IT -> using interrupts on events instead of the polling to progress the exercise

STM32 Logging Library

This library have been developed in different version.

- EventLogger: Provide logs on a MassStorage usb device plugged to the board
- EventLogger_FreeRTOS: Provide the same functionality but with the use of FreeRTOS and threads. This alternative have been made to introduce the concept of priority between the tasks present on the boards. This was needed to avoid overhead on the i/o operation needed to save the logs
- EventLogger_FreeRTOS_Flash: This is a prototype that never made it through and i'm sorry to have uploaded that yet. The idea behind this last iteration was to ensure that the mass storage device attached to the board could have been stolen. So we opted to save the log into the flash memory of the board.

each of the library contain a simple timer using the christian algorithm implemented to maintain the consistency of the log with other nodes on the network.

STM32 USB Demo

In this demo the code is partly entwined with the library. The usb support for the board required the implementation of the function used on the reception of specific usb message received on the bus and handling is left to the user. Each of the main.c present for each project in the src folder is been developed.

Part of the code present in the src/USB/\${USB_TYPE}/src is been developed/edited to ensure that the descriptor and the functioning of the device was as intended.

Iot Project Example

This projects used ionic framework on client side and Californium with Leshan on top for the server side.

Ionic provided us a simple framework to develop with javascript (or type-script) web pages that could easily translated into android/ios views.

Californium provided us a COAP implementation.

Leshan provided us an IPSO object standard representation.

Goals of the project:

Develop a mobile app wich provide listing/managing of ipso objects and simple rules (in the form [x and y and... and z] or [x and y and... and z] or.....) client side.

The expansion of the existing leshan/californium framework with the handling of the rules, their persistence, and eventually allarms as special kind of rules that needed to be notified to the client.

DandrolMobile

DandrolMobile/Src/pages contains the code developed pages used to navigate through the app.

DandrolMobile/Src/App/ipso.ts and DandrolMobile/Src/app/rule.ts contains the data model used for ipso object and for the rule we format of the rule system developed

The client is able to configure the ip of the server, view the connected ipso object, view the active rules present for the objects connected, insert new rules based on the avaiable objects.

DandrolServer

Leshan come with a simple server demo and an ipso object demo. The server demo have been expanded with the addition of a database and the rule handling.

The folders present in Iot Project Example/DandrolServer/src/main/java/org/at/dandrol/LeshanServer :

- DBManager

- alarm
- automation
- rules
- and the files `./Servlet/automationServlet` and `./Servlet/alarmServlet`

represent the code developed for the project server side.

Java Project Example

This project is the implementation of two use cases of a greater project. It uses the Hibernate layer and Swing for the panels of the two different clients. all the code present have been developed to fullfill the above mentioned use cases. The code autogenerated/edited in this case reside in the panels where the eclipse “windows builder” it’s been used

The project as whole should provide a functional multimedia resource system. The use case implemented are the “express preference” on a given multimedia resource and the “save multimedia” executed by an admin actor

Admin Client

Emulate an already logged admin that is saving a new multimedia resource on the server via swing panel

User Client

Emulate an already logged user that is saving a new preference in the system based via swing panel

Server

start the Server initializing stub resource and communicate with the clients via
`java rmi`

Jars

contains jars needed to run the project and the `.bat` with the command line arguments needed to launch the jars.