

**Title**: Thread

**Subject**: Networks for embedded systems

**Teacher**: Estephania Martinez García

**Student(s):** Sergio Eduardo Borrayo Anzaldo (ie726673)

Isaac Segovia Hernández (ie721714)

**Date**: 22/11/2023

**Theoretical framework**

**Router:** Is a node that provide joining and security services for devices trying to join the network. Routers are not designed to sleep. Routers can downgrade their functionality and become REEDs (Router-eligible End Devices).

**Leader**: The Leader is an additional role of one Router in a Thread network. The Leader is an elected role of one Router, which takes certain decisions in the Thread network such as allowing REEDs to upgrade to Routers. If the Leader of a Thread Network fails, another Router will be dynamically selected to resume the role. All Routers have the required Thread Network Data to seamlessly assume this role.

**End Device**: Is a node that communicates primarily with a single Router, does not forward packets for other network devices and can disable its transceiver to reduce power

**IPv6 addresses**: An IPv6 address is a 128-bit alphanumeric value that identifies an endpoint device in an Internet Protocol Version 6 ([IPv6](https://www.techtarget.com/searchnetworking/definition/IPv6-Internet-Protocol-Version-6)) network. The Internet Protocol ([IP](https://www.techtarget.com/searchunifiedcommunications/definition/Internet-Protocol)) is a method in which data is sent to different computers over the internet. Each network interface, or computer, on the internet will have at least one IP address that is used to uniquely identify that computer. Every device that connects to the internet is assigned an IP address. Which is why there was a concern with the number of IP addresses in IPv4, and why the Internet Engineering Task Force ([IETF](https://www.techtarget.com/whatis/definition/IETF-Internet-Engineering-Task-Force)) defined the new IPv6 standard.

**PanID**: PAN ID is the identifier of a wireless network. Thread networks are identified by three unique identifiers:

* 2-byte Personal Area Network ID (PAN ID)
* 8-byte Extended Personal Area Network ID (XPAN ID)
* A human-readable Network Name

**Development of modifications made to code**

**The git hub link:** <https://github.com/SergioB17/Practica2_Equipo5>

Network setup:

Devices and types:

-Router 1: Leader

-Router 2: End device

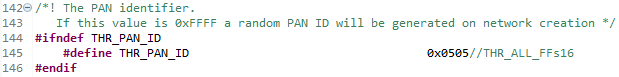
IPv6 addresses (This were randomly assigned each time the board was reprogramed):

-Router 1:

-Router 2:

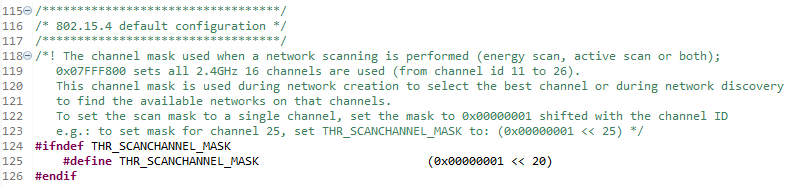
PanID (On app\_thread\_config.h):

-0x0505



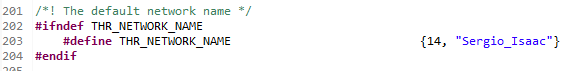
Channel (On app\_thread\_config.h):

-20

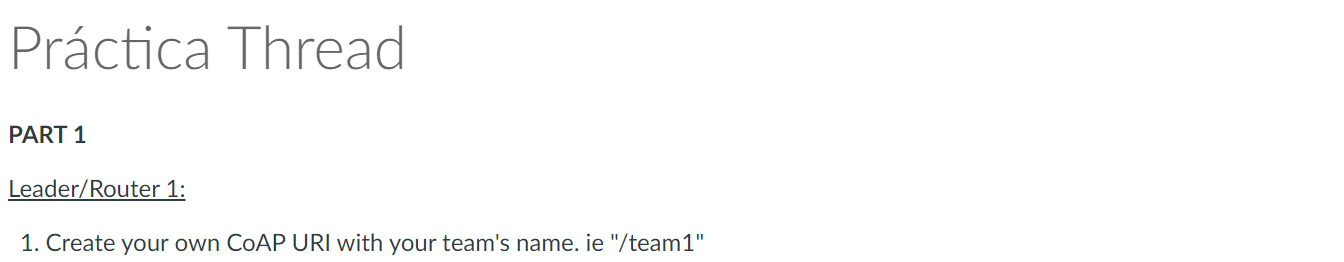


Network name:

-Sergio\_Isaac



According to the next instructions, the code was integrated sequentially. Starting with:



This step was already done with a LAB work in class, so we just use that code.

On router\_elegible\_device\_app.c we need to define the URI path names that will be used in the shell to access the resources on “router\_elegible\_device\_app.c”.



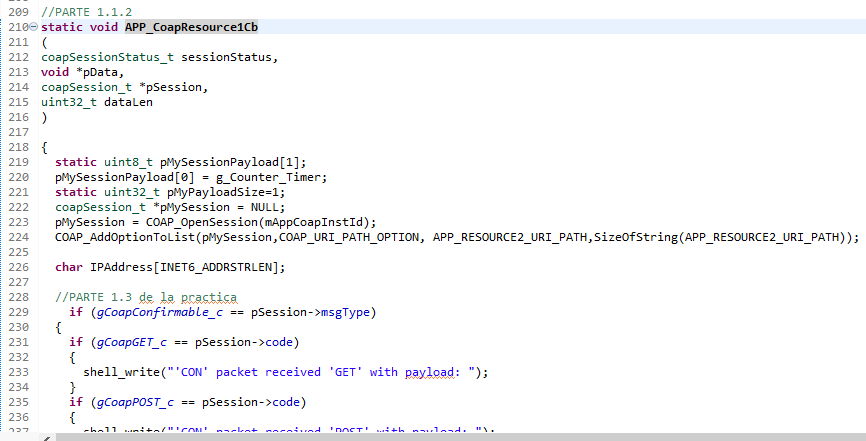
Then, we declare the URI resources with coapUriPath\_t. When using this struct the user must enter the length of the URI path and the path created in the last step on “router\_elegible\_device\_app.c”.



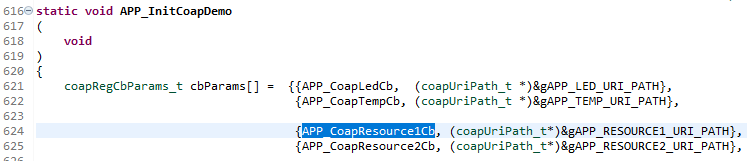
The next step consists in create the callbacks for the resources. These callbacks will handle the packet received and perform the desired action depending on the type of COAP method received on “router\_elegible\_device\_app.c”.



Now, we add the callback handler for the packet received, in this function it will be defined which action will be performed depending on the type of COAP method received on “router\_elegible\_device\_app.c”.



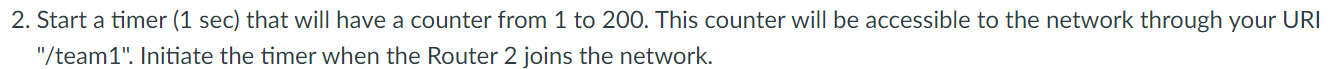
After creating the callbacks, those must be registered in the CoAP callback array in the function *APP\_InitCoapDemo(void)* on“router\_elegible\_device\_app.c”.



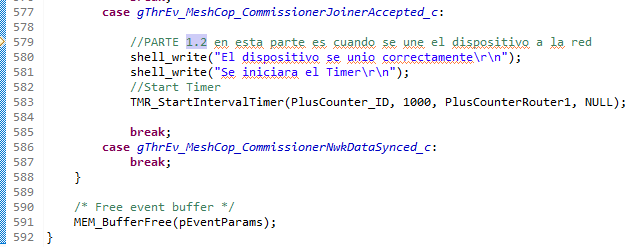
Now we need to change a setting related with the session, we need to close the session every time we open one, so we change this configuration on “coap.h”:



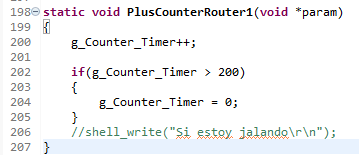
And with that, we can consider finished the 1.1 part. The next statement says:

****

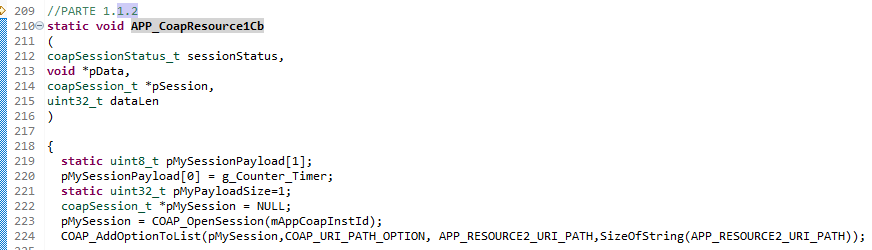
The connection is madeon “router\_elegible\_device\_app.c”, specifically on “APP\_Commissioning\_Handler” on the case where the joiner has been accepted:

****

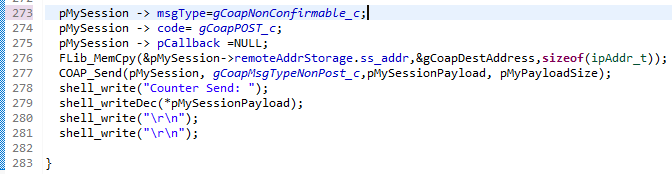
This is where the timer jumps one the counter has expired on “router\_elegible\_device\_app.c”:



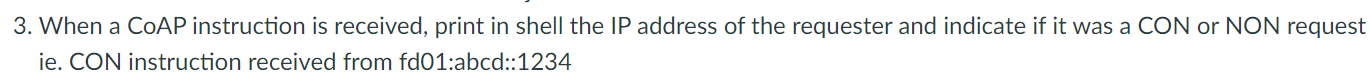
The counter is sent at the “APP\_CoapResource1Cb” function that correspond to /team5 URI. First, we assigned the counter to the Payload of the package on “router\_elegible\_device\_app.c”:



Now we send the data (we are still in the same file and function):



The next statement says:

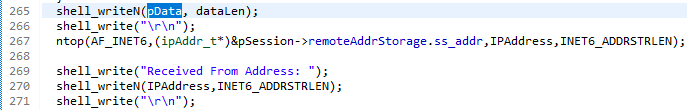


This was coded in the “APP\_CoapResource1Cb” function on “router\_elegible\_device\_app.c:





Also, the address is extracted and show from here (we are still in the same file and function)



The last statement of the first part says:



This is also coded between the last part:





**Conclusions**

**Sergio Eduardo Borrayo**:

**Isaac Segovia**: Thread es un protocolo que resulta fácil de usar y con gran flexbilidad que (parece ser) está empezando a tener gran alcance. Con respecto a la práctica, resultó sencillo implementar las funciones, pero considero que lo complicado fue el terminar de entender como se codificó originalmente para poder hacer uso de las funciones. El encontrar las partes deseadas de código para manipularlas fue lo mas tardado, al igual que el hacer la validación. Esto debido al cambio de IP cada vez que se reprogramaba la tarjeta. Los timers fueron algo muy sencillo de entender e implementar, no hubo dificultades ahí.

**References**

*How does Thread elect a Leader device?* (s/f). Internet of Things Stack Exchange. Recuperado el 22 de noviembre de 2023, de https://iot.stackexchange.com/questions/439/how-does-thread-elect-a-leader-device

(S/f). Threadgroup.org. Recuperado el 22 de noviembre de 2023, de https://www.threadgroup.org/Portals/0/documents/support/Thread%20Network%20Fundamentals\_v3.pdf