Guia 1 onion-stm32

Johnny Cubides

October 18, 2017

Onion

Comunicación Serial PC-Onion

La onion puede apuntar una terminal serial a sus periféricos UART. Para acceder a tal terminal debe tener instalado algún cliente serial. Para éste caso se hará la demostración con minicom.

Instalación de minicom

```
# sudo apt install minicom
```

Uso de Minicom

```
$ minicom -o -b 115200 -D /dev/ttyXY
```

Donde:

- **-o**: Permite reiniciar la conexión si en algún instante deja de lograrse.
- **-b**: Define la velocidad en baudios de la conexión, por default es 115200.
- **-D**: Define a qué dispositivo apuntar para la conexión el cual sera el adaptador UART.
- **X**: Puede ser visto como USB o ACM, dependerá del controlador usado.
- **Y**: Es un número que identifica al dispositivo serial en orden de conexión.

Aclaración: Es importanet fijarse de que **no haya control de hardware** en el cliente minicom sino, no podrá interactuar con el/la shell de la onion.

Para ver si hay control de hardware desde minicom hacer:

```
^A
z
o
```

Asegurarse de que esté en **NO** y guardar configuración.

Comunicación TCP

Shell segura

La comunicación permitida para la onion puede ser lograda desde un SSH (Shell segura), basta con saber el usuario y el dominio **user@domain**

Si está la onion en modo **AP** (Punto de acceso Wifi), por default tiene la siguiente dirección IP **192.168.3.1** la cual está indicando el domain. Ejemplo de uso:

```
$ ssh root@192.168.3.1
```

Aclaración: Esto conlleva a pensar que la *onion* debe tener un servidor SSH el cual es localizado como sshd en los servicios de ella misma.

Aclaración: No olvidar que si se conecta por primera vez a éste domino como ese usuario en particular debe aceptar las condiciones con la palabra completa **YES** ya que lo que hará a partir de ese momento es una negociación de llaves públicas; cabe también aclarar que las llaves públicas cambiaran cada cierto tiempo por eso esto es una shell segura.

Envio de archivos por SSH comando SCP

Este es un servicio que permitirá enviar paquetes por el protocolo orientado a la conexión y seguro TCP.

Comando:

```
$ scp archivo -r root@domain:/path_destino
```

Aclaración: En GNU/Linux un directorio (carpeta) también es un archivo, esto es adoptado de UNIX (O.S. Padre), así que puede apuntar a un directorio y con el parametro **-r** puede apuntar recursivamente a los archivos que estén contenidos allí dentro como si se tratara de un árbol.

Aclaración: Fíjese también que debe ser específico con el destino (path_destino), debe conocer con anterioridad el lugar donde alojará el archivo en la onion.

Ejemplo de uso:

```
/ChibiOS_17.6.2/rt_stm32f100rb_onion/build $
scp ch.bin root@192.168.3.1:/root/stm32f100/
```

STM32F100

ChiBios

Se trata de un sistema operativo de tiempo real (RTOS) que pone un programa intermedio para la gestión de recursos de la máquina, permitiendo hacer aplicaciones tipo semáforo y actión inmediata sobre banderas activadas.

El archivo **rt_stm32f100rb_onion.tar.gz** es descomprimido en el siguiente nivel de directorio dentro de chibios:

```
$ pwd
.../ChibiOS_17.6.2 $
tar xvf rt_stm32f100rb_onion.tar.gz
```

Al entrar allí tendremos principalmente dos comandos **Make**:

- **make all** Croscompila el chibios para el stm32f100 en **build/**, muestra los errores y advertencias si los hay.
- **make clean** Limpia el directorio build/ donde está el croscompilado **ch.bin**

JTAG

La programación es realizada a partir de **JTAG** (Joint Test Action Group), norma IEEE 1149.1. Usado como depuración de programas en submódulos embebidos.

Requiere de 6 líneas de comunicación.

- TDI (Entrada de Datos de Testeo)
- TDO (Salida de Datos de Testeo)
- TCK (Reloj de Testeo)
- TMS (Selector de Modo de Testeo)
- TRST (Reset de Testeo) es opcional.

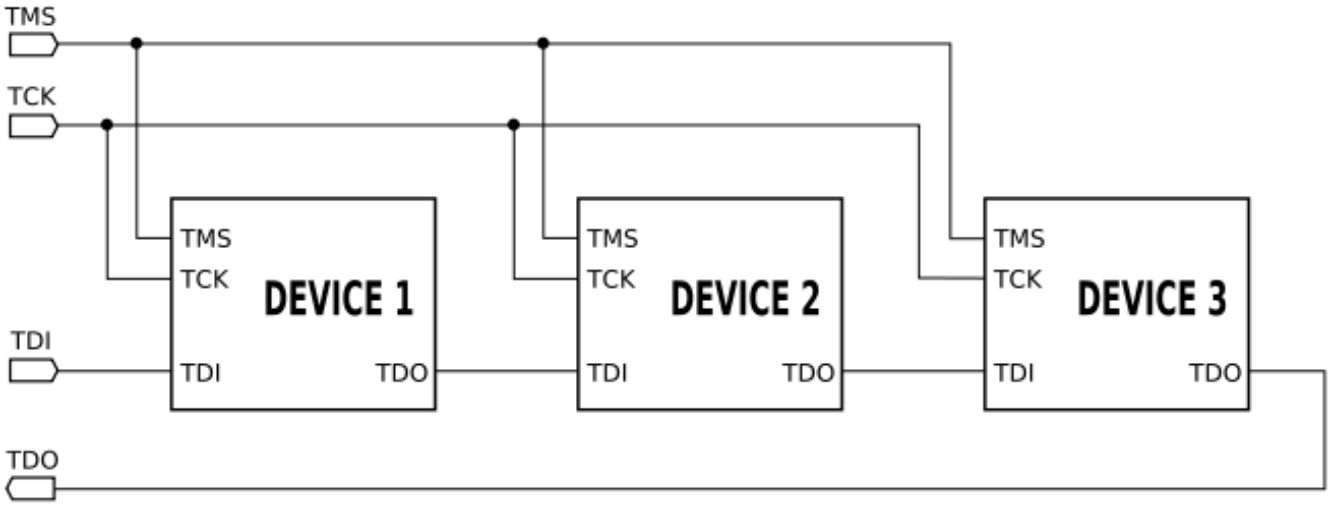

Figure 1: JTAG

Programación desde Onion por JTAG

Nota: Es bueno saber que si la placa *onion* tiene acceso a Internet no se requiere **crosscompilar** algunos paquetes, pues ya están pre-compilados en la nube. El gestor de paquetes *opkg* hará el "trabajo sucio" de verificar si tal paquete está en la red y de ser así descargarlo e instalarlo en la onion.

Se requiere tener instalado en la onion el openOCD

```
# opkg install openocd
```

Para ordenar el contenido de los programas y scripts dependientes de la programación de la memoria flash del stm32f100 basta con crear un directorio y alojar allí el siguiente contenido.

Creación de directorio en onion

```
# mkdir stm32f100/
```

Archivos que deben ser contenidos en stm32f100/

- Archivo de configuración de openocd **media__board__onion.cfg**

```
# BEGIN
source [find sysfsgpio-onion.cfg]
#telnet_port 4444
#gdb_port 333
set WORKAREASIZE 0x2000
transport select "jtag"

source [find target/stm32f1x.cfg]
reset_config  trst_only

init
reset halt
#stm32f1x unlock
#reset halt

wait_halt
sleep 10
#flash info 0
```

```
#flash probe 0
flash  write_image erase  ch.bin 0x08000000
soft_reset_halt
reset run
shutdown
# END
```

- Archivo de configuración de los pines onion **sysfsgpio-onion.cfg**

```
# BEGIN
#
# Config for using RaspberryPi's expansion header
#
# This is best used with a fast enough buffer but also
# is suitable for direct connection if the target voltage
# matches RPi's 3.3V
#
# Do not forget the GND connection, pin 6 of the expansion header.
#

interface sysfsgpio

# Each of the JTAG lines need a gpio number set: tck tms tdi tdo
# Header pin numbers: 23 22 19 21
sysfsgpio_jtag_nums 17 2 16 15

# At least one of srst or trst needs to be specified
# Header pin numbers: TRST - 26, SRST - 18
sysfsgpio_trst_num  1
sysfsgpio_srst_num  19
# END
```

Nota: Puede copiar el contenido de estos archivos si no los tiene y crear tales configuraciones en la onion.

Comando de programación

Estando dentro del directorio creado (stm32f100/) y teniedo crosscompilado el **archivo.bin** para el stm32 puede hacer uso del siguiente comando como se ejemplifica.

```
# openocd -f media_board_onion.cfg
```

Aclaración: ch.bin es el binario **croscompilado** para el stm32f10

Comunicación por UART onion-stm32

Hacer una comunicación serial entre la *onio* y el *stm32* permite por ejemplo la depuración, dar ordenes al stm32 y recibir datos para almacenar en la onion o llevar a Internet.

Como en el caso anterior (cuando se usó minicom para conectar nuestro PC a la terminal *onion*) se requiere saber cuál es el archivo representativo del hardware UART; según las especificaciones de hardware de la *onion*, se cuenta con dos de ellas: [uart0, uart1]. La **uart0** es la que usa *onion* para generar su terminal; mientras que **la uart1** es de libre uso.

Los archivos representativos del hardware se alojan en el directorio **/dev/**. Los archivos representativos según documentación para las UARTs están asociados como sigue:

- [uart0, /dev/ttyS0]
- [uart1, /dev/ttyS1]

Aclaración: Esto es posible con *sysfs* (seudo archivo del sistema de archivos virtual).

Prueba: Si desea ver el funcionamiento de la **uart1**, saque la *onion* de la placa base, genere un puente con un conector rápido (jumper) entre **Tx** y **Rx**; abra dos terminales con **SSH,** en la segunda escriba el siguiente comando:

```
cat /dev/ttyS1
```

Ahora vuelva a la primera terminal y ejecute el siguiente comando:

```
echo "Hola onion" >> /dev/ttyS1
```

Verá en la segunda terminal el mensaje transmitido por el **Tx**, retornando por **Rx** que está como sabemos "puenteado".

Depuración Serial

Aclaración: Una depuración podría hacerse con JTAG, sin embargo, no será tenido en cuenta este tema.

Para hacer depuración UART se requiere un cliente serial que permita la comunicación entre la *onion* y el *stm32*.

En éste caso se puede hacer uso tanto de minicom, como screen.

Para hacerlo por minicom puede hacer como sigue:

```
minicom -o -b xxxx -D /dev/ttyS1
```

Para hacerlo por screen puen hacerlo como sigue:

```
screen /dev/ttyS1 xxxx
```

Donde **xxxx** en ambos casos (screen y minicon) depende de la configuración que tenga el stm32 por default se esperaría **115200** baudios.

Aclaración: para la instalación de **minicom** y/o **screen** puede hacerlo como sigue:

```
opkg update
opkg install screen
opkg install minicom
```

Nota: El Anterior comando funciona sin **crosscompilar** los programas siempre y cuando se esté conectado a Internet.

Programas en la Onion que se conectan al stm32 por uart

El uso de clientes seriales, permite hacer control directo y observación de datos, sin embargo, en una aplicación estándar lo que se desea es poder enviar datos y traer datos sin la intervención del programador; Para tal fin se expondrá el siguiente ejemplo con python.

Primero se debe instalar las librerías de python para que éste sepa hacer uso de la uart1:

```
opkg update
opkg install python-serial
```

Ejemplo script de python:

```
#! /usr/bin/python
#BEGIN
# -*- coding: utf-8 -*-
# Este codigo permite leer datos desde la uart para
# Luego imprimirlo en pantalla.

import serial, time

#uart  = serial.Serial(/dev/ttyS1, 9600, timeout=1)
uart  = serial.Serial(/dev/ttyS1, 115200, timeout=1)
```

```
def main():
    while True:
        x = uart.read()
        print(x)
        #x = uart.readline() #lee hasta \n
        #print(x)

main()
#END
```