

## Fallos en proyecto Ethernet Beaglebone Black

### 1. Falla de Internet over USB

Después de confirmar que los drivers para el Ethernet (eth0) tampoco existían, el era usar la funcionalidad de "Internet over USB" (CDC-ECM) a través del puerto USB de la BBB.

Al activar las opciones de USB (CONFIG\_AM335X\_USB1 y CONFIG\_CDCECM), la compilación falló

```
LD: nuttx
arm-none-eabi-ld: /home/ubuntu/Documents/pj_BBB/nuttx/staging/libarch.a(arm_initialize.o): in function `up_initialize':
arm_initialize.c:(.text.up_initialize+0xc): undefined reference to `arm_usbinitialize'
Memory region      Used Size  Region Size  %age Used
  isram:             0 B         63 KB       0.00%
  ocm0:              0 B         48 KB       0.00%
  ddr:             273688 B       352 MB       0.07%
make[1]: *** [Makefile:217: nuttx] Error 1
make: *** [tools/Unix.mk:552: nuttx] Error 2
```

### 2. Falla de Raw Sockets (AF\_INET / SOCK\_RAW)

La segunda parte requería crear un cliente que armara manualmente los encabezados IP y UDP usando socket(AF\_INET, SOCK\_RAW, IPPROTO\_RAW).

Pero hubo un error en compilación

```
CC: my_client_raw_main.c my_client_raw_main.c: In function 'my_client_raw_main':
my_client_raw_main.c:90:36: error: 'IP_HDRINCL' undeclared (first use in this function)
  90 |     if (setsockopt(sock, IPPROTO_IP, IP_HDRINCL, val, sizeof(one)) < 0)
      |                                    ^
      |                                    |
      |                                    A not that not that not that not
```

### 3. Sockets AF\_PACKET

Nuttx sí soporta AF\_PACKET, pero usarlo para esta tarea habría requerido implementar manualmente ARP y la el header Ethernet, lo cual es más complejo. El AF\_PACKET está diseñado para interactuar con hardware de red físico como eth0 y no funciona con la interfaz de loopback para este caso, ya que obtenemos el siguiente error de protocolo no soportado:

```
Hola Cliente RAW (Nuttx)
Configuración Final: Servidor=127.0.0.1:3000, Mi IP=127.0.0.1
FATAL: Error: socket() falló. errno=93
nsh> █
```

```

Configuración: Protocolo=UDP, Puerto=3000
Servidor UDP escuchando en INADDR_ANY:3000. Esperando mensaje...
raw_client
Iniciando raw_client
Interfaz 'lo' tiene índice 1
Socket AF_PACKET creado y ligado a 'lo'.

hola
Mensaje (escriba 'exit' para salir): Enviando paquete completo (46 bytes) a la interfaz lo...
ERROR: sendto falló: 97

1+1
Mensaje (escriba 'exit' para salir): Enviando paquete completo (45 bytes) a la interfaz lo...
ERROR: sendto falló: 97

exit
Mensaje (escriba 'exit' para salir): Cerrando socket.
nsh>

```

#### 4. AF\_PACKET con SOCK\_RAW

ERROR 97, este error sale cuando se intenta enviar un paquete UDP más grande que el MTU (Maximum Transmission Unit) de la red. Pero en este caso, estaba enviando 46 bytes. Probablemente porque el header manual no cumplía con algún requisito estricto de alineación o formato que el driver interno necesita.

```

hola
Mensaje (EXIT para salir):
[+] --- PAQUETE CONSTRUIDO (AF_PACKET) ---
  [ETHERNET] Src: C4:DE:E2:C0:A8:9C -> Dst: FF:FF:FF:FF:FF:FF
  [IP]       Src: 192.168.1.28 -> Dst: 192.168.1.28 (Proto: 17)
  [UDP]      Src Port: 12345 -> Dst Port: 3000
Intentando enviar 46 bytes...
Error enviando: 97

```