

Instalación y Configuración de Sistemas Embebidos en una SBC Intel Galileo 2 como Base de Plataforma IOT

César Omar Aranda

IISCOT/ FCSyA, Universidad del Aconcagua, Catamarca 147, Mendoza, Argentina
Fac. de Ingeniería, Universidad de Mendoza, Boulogne Sur Mer 683, Mendoza, Argentina
cesar.aranda@uda.edu.ar

Resumen. Este trabajo describe cómo convertir una Impresora 3D de escritorio estándar en una impresora inteligente, esto es que pueda integrarse a un ecosistema de IoT. Considera el uso de una SBC del tipo Galileo 2 Gen, desarrollada por Intel. Resume el análisis y la evaluación para diferentes distribuciones del sistema operativo Linux. Detalla los pasos seguidos en la instalación y configuración del mismo para lograr un sistema relativamente estable y también del conjunto de aplicaciones básicas capaces de proveer diferentes servicios (comunicación, base de datos y lenguajes de programación) que permitan realizar el monitoreo y control remotos de una impresora 3D conectada a la placa elegida.

Palabras Clave: IOT, SBC, Sistemas embebidos, Sistema operativo Linux, Impresora 3D inteligente, Galileo 2 Gen.

1 Introducción

Este trabajo surge como parte de dos proyectos mayores, de investigación y desarrollo. El primero en el ámbito de tecnologías aplicadas a los ambientes de IOT (en la Universidad del Aconcagua) y el segundo en el marco de una tesis en una carrera de postgrado, para optar al título de Magister en Teleinformática (Universidad de mendoza), donde uno de los supuestos es que resulta viable usar una placa SBC Intel Galileo 2 como Host de una impresora 3D, para convertirla en un dispositivo de Internet de las Cosas (IoT). La mencionada placa ingresó al mercado argentino, con una popularidad y aceptación mucho menor que las Raspberry y, al momento de hacer el análisis incluido en este informe, presentaba amenazas de abandono por parte de Intel.

Una SBC (Single Board Computer) es una PC de placa única. Esto quiere decir que a diferencia de las computadoras tradicionales, las SBC son placas electrónicas que contienen todos (o la mayor parte de los componentes) de una computadora. Dos de las características que la hacen aplicable al problema en estudio son el menor precio comparado con una PC y el reducido tamaño

Los fundamentos de la elección de la mencionada placa, se encuentran en los anuncios realizados por el fabricante, indicando que las placas Intel (en cualquiera de sus modelos: Galileo, Edison, Joule o Curie) han sido pensadas para ser usadas en proyectos de IoT, así como en informes de revisión técnica realizados sobre la misma. Como afirma Nayyar, “Intel Galileo se puede utilizar para hacer diferentes tipos de

proyectos en términos de Domótica inteligente, Robótica, Drones e incluso se puede utilizar para la implementación exitosa de proyectos de Smart Cities” [1]

Al buscar y analizar proyectos que se encuentren terminados y disponibles en Internet o en libros publicados, que tengan relación directa con el objetivo de este trabajo y que usen la SBC Galileo, se detectan: un desarrollo de firmware para control directo de una impresora 3D [2] y el de control de un CNC [3]. En ambos, el software está parcialmente terminado y usan MS Windows como sistema operativo. Un tercer proyecto sigue la misma línea conceptual de este trabajo al usar una SBC Intel Edison para controlar los procesos de impresión 3D [4]. Muchos otros proyectos se basan en placas Raspberry, pero ninguno de ellos usa la placa elegida.

1.1 Internet de las Cosas

IoT es un concepto aparecido en 1999 que se refiere a la interconexión digital de objetos cotidianos (cosas) con internet. Resulta de la convergencia de tecnologías de conectividad inalámbrica, sistemas micro electromecánicos (MEMS) e Internet.

Usualmente los objetos con soporte para comunicación de máquina a máquina (M2M) reciben la denominación de dispositivos inteligentes (smart devices). Cuando esta comunicación se realiza usando internet, pasan a formar parte de IoT.

IoT asume que las “cosas” deben ser identificables de manera única, capaces de recopilar datos de sensores, de comunicarse y transferir datos a través de una red. Dichos datos podrían ser utilizados para propósitos de monitoreo, procesos big data, o incluso para controlar a esa misma “cosa”. [5]

Un Sistema o solución IoT puede verse, simplemente, como un conjunto de dispositivos diseñados para producir, consumir o presentar datos sobre uno o más eventos u observaciones. Este puede incluir dispositivos que generan datos como un sensor, dispositivos que combinan datos para deducir algo, dispositivos o servicios diseñados para clasificar y almacenar datos, o también para presentar los datos. Uno de ellos, o todos, pueden estar conectados a Internet.

Una solución de IoT puede incluir una o varias de las cualidades anteriores cuando se combinan en un único dispositivo. Son los casos de una cámara web, o bien una unidad de monitoreo junto a un conjunto de sensores de una estación meteorológica, o bien un sistema complejo de sensores dedicados, almacenamiento de datos y presentación dentro de un sistema de automatización domótica.

1.2 Escenario de aplicación

Desde hace algunos años se observa el incremento de la actividad relacionada a la impresión 3D, debido a la reducción del tamaño de equipos, a la disminución de costos de los materiales involucrados y a la popularidad de movimientos como Maker o DIY.

El problema de las actuales impresoras 3D de escritorio es que poseen una funcionalidad y controles de proceso limitados, no desde el punto de vista del tamaño o calidad de las piezas obtenidas, sino fundamentalmente desde su autonomía de trabajo o de la mejora del proceso de impresión.

Las impresoras adquiridas o construidas pertenecientes a la gama estándar dependen de una computadora o notebook conectada para realizar las tareas de administración usando software especializado, particularmente mientras dure la impresión. Este software suele denominarse Aplicación de Host de Impresión. Estas impresoras, en general, no permiten el monitoreo o control remoto. En especial, dentro del universo de las impresoras DIY o que no poseen un respaldo comercial significativo

Una forma de obtener las funcionalidades y ventajas mencionadas es integrar a la impresora a una la placa SBC y convertirla en un dispositivo “inteligente”.

1.3 Generalidades de la SBC Galileo Gen 2

La Galileo es una placa de desarrollo, basada en el microprocesador Intel Quark SoC X1000. Con un conjunto de instrucciones similares al procesador Intel Pentium de 32 bits, con un solo núcleo funcionando a velocidades de hasta 400 MHz. [6]

La placa ofrece compatibilidad con diferentes interfaces de E/S estándar en la industria, a través de una ranura mini-PCI Express, un puerto RJ45 Ethernet de 100 Mb, conectores USB 2.0 para host (tipo A estándar) y para cliente (tipo B micro USB).

Adicionalmente posee conjuntos de pines para FTDI, UART, ICSP, SDIO, y pines independientes que proveen entradas/salidas digitales, entradas analógicas, salidas PWM de 12 bits de resolución, 1 SPI master, I2C master.

A nivel de memoria presenta 256 MB DDR3, SRAM de 512 kb integrada, Flash NOR de 8 MB (para firmware y bootloader), EEPROM de 8 KB, y una ranura para tarjeta microSD de hasta 32 GB.

Tiene RTC integrado y compatibilidad de hardware y pines con Arduino Uno R3.

2 Infraestructura de red

Para lograr que esta impresora pase a formar parte de un ámbito de IoT, es necesario modelar la misma como parte de una red informática.

En dicha red deberíamos encontrar los siguientes componentes:

- Host Server de impresión, conectado a la electrónica de control de la impresora 3D, con capacidad de actuar además como Servidor Web Dedicado,
- Una impresora 3D (o más), identificada como dispositivo esclavo, que pueda comunicarse con el Host usando protocolos de bajo nivel.
- Un dispositivo (o más) que pueda actuar como Cliente Web para acceder a una interfaz gráfica de usuario, de monitoreo y/o control. Eventualmente con capacidad para realizar tareas de administración sobre el Host.
- Router que permita conexiones cableadas y/o inalámbricas para interconectar los nodos anteriores. Opcionalmente, ser el puente de enlace con Internet.

En la Fig. 1 pueden apreciarse los 4 nodos mencionados de la red prototipo.

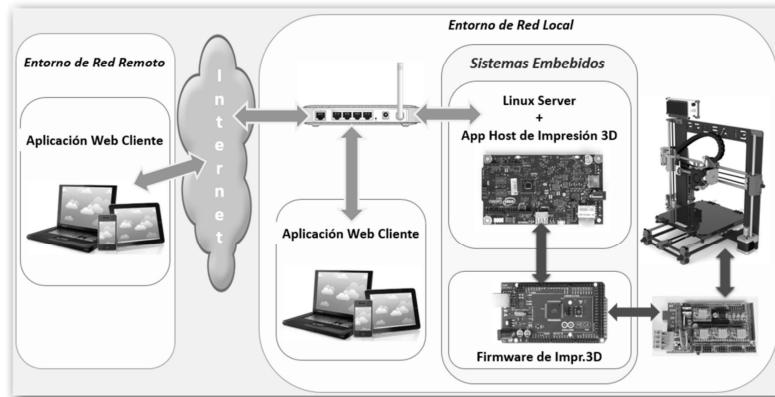


Fig. 1. Esquema de componentes del ecosistema IOT para impresión 3D.

2.1 Consideraciones iniciales

Una de las particularidades del trabajo con placas SBC es que requieren manipular sistemas operativos embebidos.

Si bien, en general, las tareas de instalación y configuración de un SO de servidor son similares al trabajo sobre una PC tradicional, no resultan triviales al momento de preparar el sistema embebido para una solución específica. Por ello, en este trabajo se realizará la descripción de los pasos seguidos para obtener un Server Host operativo sobre la Intel Galileo Gen 2.

La placa elegida admite desarrollos bajo diferentes sistemas operativos.

Dado que se trata de una elección preferente de los desarrolladores de SBCs, en particular de movimientos Makers, debido a la gratuidad y a la disponibilidad del código fuente, se adopta como sistema operativo una distribución basada en Linux.

Cabe destacar que, aunque existe una comunidad grande atrás de esta tecnología que puede colaborar en los desarrollos, durante las pruebas se detectaron complejidades debidas a un bug operacional, con referencias fueron muy difíciles de hallar.

Los componentes de software embebidos se pueden asociar a 5 capas dependientes.

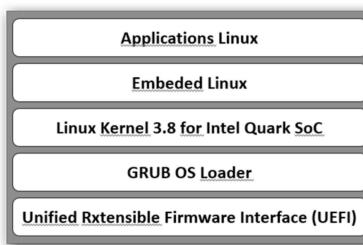


Fig. 2. Capas de Software Estándar Embebido en la SBC.

El primer paso para asegurarse que el software sea funcional en la placa elegida, Galileo de 2^a Generación, es actualizar su firmware. Algunos errores que se produjeron durante las primeras pruebas fueron consecuencia de trabajar sobre la placa existente, que disponía de un firmware antiguo (la versión 1.0.2 preinstalada en fábrica).

2.2 Alternativas de selección

El sistema operativo Linux puede embeberse en una SBC Intel Galileo a partir de imágenes preparadas según 4 versiones fundamentales:

Versión A: "SPI-Image". Es un Kernel mínimo de Linux (micro Yocto de 766kB) adaptado para alojarse en la memoria flash de 8MB de la placa Galileo. Preinstalada en las placas provistas por Intel. Permite usar al dispositivo como si se tratara de un Arduino Uno R3. No necesita de tarjeta SD para arrancar. El sistema comienza apenas se energiza la placa y tiene funcionalidad limitada. Por ejemplo, no tiene drivers WiFi y los sketches Arduino no son persistentes, es decir el código que se cargue a la placa se pierde al apagarla.

Versión B: "SD-Card Image". Denominada distribución Linux Yocto 1.4 Poky, es una imagen suministrada por Intel para ser alojada en una memoria auxiliar SD, que habilita el uso de Wifi. La última versión es la 1.1.1 (14/Nov/2016).

Versión C: "IoT Devkit Image". Es una imagen ampliada desde la anterior. Facilita la conexión a redes, posee más herramientas de desarrollo y soporte de OpenCV para python. Es una versión basada en Linux Debian con algunas herramientas básicas instaladas, que mejoran las posibilidades de desarrollo y uso. Asume requerimientos genéricos de la placa para ser usada en proyectos de IOT, incorporando drivers WiFi (all Intel-chipset WiFi cards), intérprete python, Node.js, SSH (Secure Shell), OpenCV (open-source computer vision application), ALSA (Advanced Linux Sound Architectures), V4L2 (Video4Linux2). Como el caso anterior, requiere de memoria microSD auxiliar.

Versión D: Imagen "Personalizada". Cualquiera de ellas requiere de memoria microSD auxiliar. Cada imagen ha sido preparada por uno o más desarrolladores en función de necesidades específicas y con criterios diferentes, usando el kernel Linux mínimo para Galileo provisto por Intel junto a un sistema estándar. Se encuentran:

Versión D-1: Galileo-Debian, SD_Card. Imagen generada por el proyecto del mismo nombre en Sourceforge.net. Utiliza la adaptación del núcleo de Linux 3.8.7 de Intel y monta encima un sistema básico Debian GNU/Linux 7 (Wheezy) estándar para i386.

Versión D-2: Galileo-Debian, Igor-SDK_and_SD_Card. Imagen similar a la anterior, resultado del mismo proyecto. Se modifican los sistemas booteables y por ende las opciones de arranque de GRUB. Existe un usuario user además de root, con menores privilegios que éste último y se incluye un SDK formado por herramientas de programación en perl, python y c++.

Versión D-3: Yet Another Debian, Comunidad Intel. Imagen (YAD4galileo1.img.zip) que parte de la "IOT devkit Image", y ha reemplazado Yocto por Debian Wheezy con el ajuste usando el parche para libc6 incluyendo muchas herramientas como mc, dselect, vncserver, etc. El sistema permite bootear la placa Galileo vía ssh o telnet sobre ethernet sin necesidad de usar una conexión serie rs-232.

Versión D-4: AjanthaB. Versión de Linux Yocto para Galileo gen2 con OpenCV preinstalado. Es una versión similar a SD_card_image.

Versión D-5: Ubilinux. Versión provista por Emutex Labs. Se basa en las fuentes oficiales del núcleo para las CPU Intel Quark y construida a partir de una instalación mínima de Debian. Incluye controladores para WiFi y otros periféricos y permite el acceso a los repositorios de Debian, con repositorios adicionales de dicha compañía, y los archivos necesarios para bootear (UEFI) en placas Intel (especialmente Edison).

Tabla 1. Otras características de interés de las imágenes “personalizadas”

Versión	Sistema de Archivos	Tamaño Instalado	Año	Repositorio Estándar
D-1	Debian Wheezy	~ 900 MB	2014	Sí
D-2	Debian Wheezy	~ 2 GB	2014	Sí
D-3	Debian Wheezy	~ 4 GB	2014	Sí
D-4	Yocto	~ 320 MB	2015	No
D-5	Debian Jessie	~ 3 GB	2015	Sí /parcial

2.3 Pruebas realizadas y resultados

Todas las imágenes descargadas, arrojaron errores (segfault) durante su funcionamiento cuando se intenta conectar a la placa vía ssh.

Al analizar en detalle el error producido se detecta que se trata de una violación de segmento en el módulo libpthread perteneciente al paquete libc6. La violación de segmento, en general sucede cuando un programa intenta acceder a una zona de memoria a la cual no tiene permiso.

En la investigación se detecta que si bien el tema ha sido planteado, no hay una solución oficial definitiva. Tampoco se aclara si se trata de un error del microprocesador Intel Quark o del kernel del sistema operativo Linux. Aunque se sospecha que es debido a una llamada a una instrucción de microprocesador 586, cuando el Quark es básicamente un 486 mejorado.

Para la Galileo 2 existe una referencia de posible solución en una de las comunidades de Intel consultadas, que propone recompilar el kernel con opciones de compilación ajustadas o usar el parche realizado para la versión Galileo 1, no probado oficialmente. Dado que el archivo del parche no está disponible para su descarga directa, se decide extraer de la imagen de la versión D-3 el archivo en conflicto y copiarlo (sin recompilar) en el sistema de archivos de la imagen personalizada para la versión Galileo 2 que se prepara en este trabajo.

Se procede a instalar, parchear y probar cada una de las imágenes disponibles.

Las pruebas realizadas indican que, aparentemente, se resuelve el problema, al menos para cada una de versiones instaladas.

En las imágenes de las versiones A y B, con Linux Yocto, se observa que el sistema operativo final carece de las aplicaciones y herramientas requeridas para este proyecto, además del hecho de no disponer de soporte a otros paquetes a través de repositorios como ampliamente lo tiene una distribución como Debian o Ubuntu. Se considera que se encuentran incompletas al no contar con herramientas que permitan administrar de manera simple un servidor Linux.

En la imagen de la versión C, también con una versión de Linux Yocto, se observa que el sistema operativo final si bien con herramientas básicas tampoco cuenta con herramientas de administración que faciliten las tareas de instalación o mantenimiento del sistema, además de contar con aplicaciones preinstaladas que no resultan útiles a este proyecto en particular.

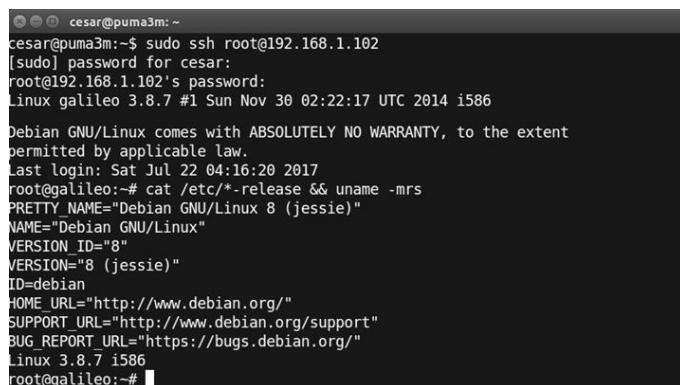
En este trabajo se necesitan servicios para satisfacer operaciones remotas de configuración y monitoreo, tanto ligadas directamente a la placa Intel en cuestión como de otros dispositivos conectados de la red.

Por lo anterior, y atendiendo a las tareas de instalación, configuración y mantenimiento del sistema operativo y de las aplicaciones de servidor que deberán funcionar en el host, se decide trabajar con alguna de las versiones del tipo D.

En primer lugar, se descarta la alternativa D-5 (Ubilinux) porque no fue posible obtener su imagen durante las primeras pruebas para el modelo Galileo. Se encuentra disponible el archivo para la placa Intel Edison pero se entiende que a pesar de tener el mismo procesador Quark, posee otras características (como la GPU) que dan lugar a un kernel diferente por lo que se decide no usarla. Se espera a la aparición de una nueva versión de esta distribución.

Luego de las pruebas iniciales con las imágenes disponibles, se decide trabajar en base a la opción D-2. En caso que, durante el avance de este trabajo con la versión elegida, se produjeran nuevos inconvenientes y no pudieran resolverse de manera rápida, se probará la próxima versión Ubilinux cuando esté disponible.

Dado que la mayoría de las distribuciones y versiones lo hacen basándose en Linux Debian 7, se prueba de migrar a una versión posterior a partir de algunas existentes.



```
cesar@puma3m:~$ sudo ssh root@192.168.1.102
[sudo] password for cesar:
root@192.168.1.102's password:
Linux galileo 3.8.7 #1 Sun Nov 30 02:22:17 UTC 2014 i586
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jul 22 04:16:20 2017
root@galileo:~# cat /etc/*-release && uname -mrs
PRETTY_NAME="Debian GNU/Linux 8 (jessie)"
NAME="Debian GNU/Linux"
VERSION_ID="8"
VERSION="8 (jessie)"
ID=debian
HOME_URL="http://www.debian.org/"
SUPPORT_URL="http://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
Linux 3.8.7 i586
root@galileo:~#
```

Fig. 3. Reporte de versión Linux Debian 8 (Jessie) instalada.

Se observa que, si bien el sistema inicia normalmente y permite trabajar en general, nuevamente aparece el bug (recientemente parcheado) al conectar mediante ssh.

La decisión final, entonces, es preparar el servidor Linux y configurar el Host de impresión con las aplicaciones requeridas en este proyecto, a partir de la versión D-2 (Igor-SDK_and_SD_Card) la cual posee herramientas de administración básicas, sistema de archivos estándar, con menos opciones de arranque desde Grub que el resto, y que permite preparar el sistema operativo de un servidor completo usando el soporte a los diferentes paquetes a través de repositorios estándares.

3 Instalación y configuración (síntesis de pasos)

Para instalar el sistema operativo Linux que servirá de plataforma principal, se respetan las directivas dadas por Intel [7] y otras recomendaciones disponibles en la web, partiendo del archivo elegido, con la imagen descargada desde el repositorio del proyecto Galileo Debian.

3.1 Sistema operativo básico

- Se descomprime el archivo descargado y se transfiere a una tarjeta de memoria micro SD (FAT32). Dado que la imagen ha sido preparada para un disco de 2GB, en caso que la memoria posea mayor capacidad es conveniente ampliarla dejando, aproximadamente, un 10% del total sin asignar.
- Se inserta en la Intel Galileo Gen 2, se conecta y enciende la placa.
- A los efectos de observar el reporte de arranque del sistema y poder hacer el primer acceso al mismo se usa una conexión serie (115200 / 8 / 1 / N) y una consola cliente provista por putty (en futuros accesos se puede usar ssh).
- Para acceder al sistema, los datos usuario/clave por defecto son root/root.
- Se detiene la interfaz de red y se modifican los archivos interfaces, resolv.conf, hosts y hostname, ajustando los parámetros a la red donde va a trabajar. Se fija una IP estática al host (192.168.1.99), que será el punto de vinculación con la impresora “inteligente”.
- Se activa la interfaz de red

3.2 Actualización de Debian 7 (Wheezy)

- Dado que existen restricciones de versión en el uso de la librería libc6, deben indicarse al momento de actualizar el sistema operativo Linux que servirá de plataforma en el host servidor. Se crean o actualizan los archivos preferences y source-list ubicados en /etc/apt, con las restricciones y las direcciones de los repositorios Debian
- Se realiza la actualización de paquetes (realizado con una demora importante debido a la velocidad de descarga en la red y por la cantidad de cambios sufridos por el sistema Debian), del sistema desde la versión Wheezy 7.6 a la última 7.11
- Como se indicó anteriormente, esta imagen produce errores de fallo de segmento. Por lo que se procede a aplicar el parche encontrado. Se extrae de la imagen del sistema denominado YAD4galileo1.img.zip ella el archivo del módulo libpthread-2.13.so.so, se crea el enlace simbólico requerido y se copian en el directorio correspondiente de la imagen personalizada de este trabajo.

3.3 Instalación y configuración de las aplicaciones de servidor y Host

La aplicación de Host de Impresión elegida (Printron), requiere de Python en sus versiones 2.6.x o 2.7.x. No se realizan cambios, ya que el sistema recién montado tiene preinstalada la 2.7.3

- Se instala el servidor de bases de datos MySQL y el driver PyMySQL
- Se instala y se configura Lighttpd como servidor Web.
- Se crean la base de datos y los scripts para verificar el funcionamiento conjunto.
- Se deben crear los archivos .csr (con los detalles del dominio) y .key (con la clave), colocándolos en un directorio apropiado. Se crea un archivo de certificado autofirmado (.crt) y un archivo de la combinación (.pem) y se ajusta la configuración de Lighttpd para usar el mismo.
- Se instala Printron (basada en Python y formada por printcore, prонsole y prонterface) como aplicación principal del host de impresión 3D de manera completa conforme a lo aconsejados por el autor.
- Se integra la herramienta de laminado denominada Slic3r y se parametriza la aplicación Printron para que la reconozca.
- Se instalan y configuran OpenCV [8], FSWebcam y Mjpeg_streamer.
- Se conecta la placa Mega 2560 encargada del control directo de la impresora 3D y se la cámara web al puerto USB Host de la SBC (se usa un hub USB para ello).
- Se realizan pruebas de comunicación, funcionamiento y control de la impresora así como de captura y transmisión de video [9], independientes y conjuntas, todas accediendo de manera remota vía ssh [10] o http.

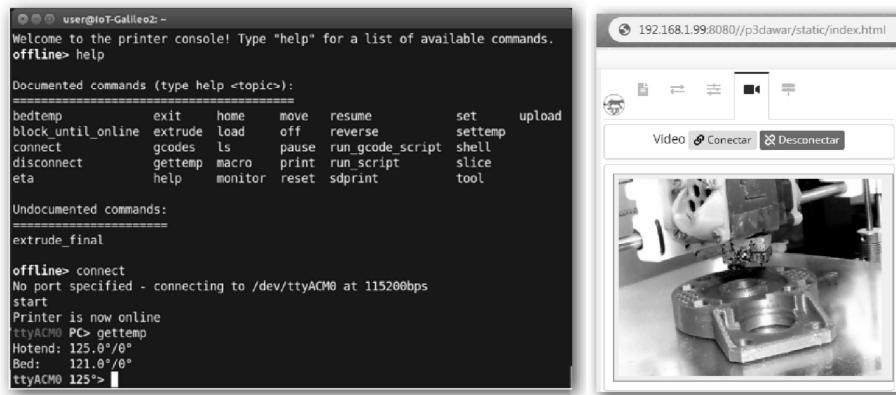


Fig. 4. Pruebas de control de impresora 3D (izq.) y web streaming (der.).

4 Conclusiones

Se comprende que Intel haya decidido discontinuar la familia de placas basadas en el procesador Quark SoC X1000 con arquitectura de 32 bits, a favor de otras con

procesadores de 64 bits, en gran medida por las grandes dificultades de administrar software sobre ellas y a las limitaciones dadas por su velocidad de trabajo.

Dada la complejidad para resolver los fallos operativos, y las dificultades para configurar o usar software actualizado, no se aconseja su uso con fines educativos.

Sin embargo, la SBC Galileo es una placa con prestaciones aceptables para satisfacer el objetivo de convertir a una impresora 3D aislada en un dispositivo inteligente para ser usado en entornos de IoT.

Es claro que, para dar soluciones de IoT en el futuro se evalúen para la elección SBCs de diferentes fabricantes que se encuentren disponibles y vigentes en ese momento.

En síntesis, en caso de disponer de una placa Intel Galileo Gen 2, ella sigue siendo vigente para proyectos de IOT, en los que no se requiera de capacidades de cómputo importantes.

Referencias

1. Nayyar, A.; Puri, E.V.: A Review of Intel Galileo Development Board's Technology. *Journal of Engineering Research and Applications*, Vol. 6, Issue 3, (Part - 4), pp.34-39. (2016). https://www.researchgate.net/publication/305671432_A_Review_of_Intel_Galileo_Development_Board's_Technology. Accedido en Mayo de 2017
2. Casper, P.: *Galileo-3D-Printer-Firmware*. Documento README.md p. 1. <https://github.com/vadlak/Galileo-3D-Printer-Firmware> (2015). Accedido en Mayo de 2017.
3. Scull, B.: *Intel Galileo Controlled CNC Plotter*. pp. 1-8. <https://www.instructables.com/id/Intel-Galileo-Controlled-Cnc-Plotter/> (2016). Accedido en Mayo de 2017.
4. Elgert, M.D.; Djedid, B.; Elgert, J.; Lindvall, K.: *3D Controller Bot, Intel Edison expands a new 3-D printer market*. pp. 1-5. <https://www.hackster.io/4378/3d-controller-bot-8e0ee1> (2015). Accedido en Mayo de 2017.
5. De Souza, M.: *Internet of Things with Intel Galileo*. Packt Publishing, Birmingham, pp.1-2 (2015)
6. Intel: *Doc 22795(ON329681-003US)-Datasheet Intel Galileo Gen 2 Development Board*. pp. 1-5. <https://communities.intel.com/docs/DOC-22795> (2014). Accedido en Junio de 2017.
7. Intel: *Intel Galileo and Intel Galileo Gen 2, Getting Started Guide*. Rev 008 y Rev 007US. https://communities.intel.com/servlet/JiveServlet/download/252925-83473/Galileo_GettingStarted_329685_007.pdf (2015). Accedido en Junio de 2017
8. Ramon, M.C.: *Intel Galileo and Intel Galileo Gen 2 - API Features and Arduino Projects for Linux Programmers*. ApressOpen/Apress Media, pp.319-399 (2014)
9. Dundar, O.: *Home Automation with Intel Galileo*. Packt Publishing, Birmingham, pp.105-123 (2015)
10. Schwartz, M.: *Intel Galileo Networking Cookbook*. Packt Publishing, Birmingham, pp. 9-11 (2015)