 <b>CONTINEA</b> <small>Microprocesamiento modular + Conectividad</small>	<b>Embedded Linux 5.2</b>	Nota de Aplicación CoTC-007
	<b>Digi CC9C + Display VGA color</b>	Publicado: 00/00/0000 Página 1 de 17

Revisión	Fecha	Comentario	Autor
0	04/04/2011	Prueba de Digi Embedded Linux 5.2	Ulises Bigliati

## Índice

<b>Introducción</b> .....	<b>1</b>
<b>Algunas mejoras</b> .....	<b>2</b>
La documentación .....	2
Más hardware soportado .....	2
Mas drivers de dispositivos .....	2
Nuevas funcionalidades en el IDE .....	2
Mas utilidades precompiladas .....	2
OpenNTPD .....	2
Python .....	3
Net-SNMP .....	3
SQLite .....	3
ALSA utils .....	3
Ethtool .....	3
<b>La instalación</b> .....	<b>3</b>
<b>Los proyectos</b> .....	<b>4</b>
<b>Bibliotecas de funciones (library)</b> .....	<b>4</b>
<b>Libraries estáticas vs. dinámicas (shared)</b> .....	<b>4</b>
<b>Creación de un proyecto de biblioteca de funciones (Library)</b> .....	<b>5</b>
<b>Aplicaciones gráficas con QTDesigner</b> .....	<b>6</b>
Creación del proyecto .....	6
La perspectiva Qt .....	7
Ubicar componentes en el formulario .....	7
Escribir código para la aplicación Qt .....	8
Compilar y correr la aplicación .....	9
<b>Conclusiones</b> .....	<b>9</b>

## Introducción

Evaluamos la versión 5.2 del kit de desarrollo Digi Embedded Linux (DEL) sobre la base de la nota CoAN-020 ya que básicamente estaremos repitiendo los pasos de aquel trabajo pero con enfoque en las mejoras detectadas.

Nuevamente utilizamos el módulo ConnectCore 9C<sup>1</sup> junto a su kit de desarrollo al cual le agregamos un display LCD de 640x480.


En esta versión de Linux Embedded, se incorporan varias mejoras con respecto a la versión 4.0 examinada en la nota CoAN-020. DEL5.2 viene con Kubuntu 8.10 y kernel 2.6.27.11<sup>2</sup> contra Kubuntu 6.06 y kernel 2.6.17. Y una ventaja notable es la posibilidad de ejecutar fluidamente el entorno de desarrollo sobre una máquina virtual, lo cual no era tan fácil en la versión anterior ya que no era posible la instalación de utilidades de la máquina virtual<sup>3</sup> fundamentales para lograr un uso eficiente de las interfaces de usuario y del hardware de la máquina anfitrión. Por lo tanto, recomendamos una vez más la instalación de DEL5.2 en modo completo, esto es, la distribución de Linux + IDE<sup>4</sup>.

<sup>1</sup> <http://www.digi.com/support/productdetl.jsp?pid=2863&osvid=0&s=227&tp=1>

<sup>2</sup> La versión 4.0 venía con Kubuntu 6.06 y versión del kernel de Linux es la 2.6.17.

<sup>3</sup> Las llamadas "Guest additions" de Virtual Box. ([www.virtualbox.org](http://www.virtualbox.org))

<sup>4</sup> Las herramientas de desarrollo consisten en el framework Eclipse, desde el cual se pueden desarrollar proyectos que involucren la compilación del kernel de linux, módulos cargables, el bootloader, del sistema de archivos, y por supuesto, las aplicaciones de usuario. Cabe destacar que el Embedded Linux que correrá en los módulo de Digi no se corresponde con ninguna de las distribuciones más populares, sino que se trata de un trabajo del fabricante para portar el kernel de

	<b>Embedded Linux 5.2</b>	Nota de Aplicación CoTC-007
	<b>Digi CC9C + Display VGA color</b>	Publicado: 00/00/0000 Página 2 de 17

## Algunas mejoras

### La documentación

Hay nuevos documentos que cubre aspectos no abordados en versiones anteriores, algunos documentos ya existentes fueron ampliados y reorganizados.

- Digi ESP™ for Embedded Linux *New*
  - Manual de usuario del entorno DEL5.2 en formato HTML mucho más completo y ordenado que el anterior.
- Building Your First Application.pdf
  - El mismo documento presente en versiones anteriores, útil para una familiarización rápida con el entorno de desarrollo.
- U-Boot Reference Manual
  - Documento genérico, independiente del entorno DEL, también presente en versiones anteriores
- Customizing platform code.pdf *New*
  - Guía práctica para la generación de código de soporte para la creación de plataformas de hardware propias. Esto es, en reemplazo de las plataformas de hardware estándar del fabricante, correspondientes a las placas de prototipos de los kits. Así, desde el IDE es posible generar un kernel de linux adaptado a las necesidades específicas del desarrollador.
- Digi Embedded Linux. Welcome
  - Instructivo con información relativa a la instalación.

### Más hardware soportado

En esta versión de DEL están soportados los siguientes módulos:

- ConnectCore 9C / Wi-9C
- ConnectCore 9P 9360
- ConnectCore 9P / Wi-9P 9215
- ConnectCore 9215
- Connect ME 9210
- ConnectCore 9M / Wi-9M 2443

### Mas drivers de dispositivos

Para incluir en el kernel, en parte acompañando los requerimientos de los nuevos módulos, se han adicionado drivers para ADC, PWM, los FIMs, sonido, 1-wire, USB host/device, etc.

### Nuevas funcionalidades en el IDE

Se agrega una perspectiva de Eclipse para el desarrollo de aplicaciones con interfaz gráfica de usuario mediante Trolltech's Qt Designer . Una interfaz gráfica para el desarrollo de aplicaciones GUI con C++.

### Mas utilidades precompiladas


Estas utilidades son las aplicaciones de usuario que pueden incluirse en el sistema de archivos de nuestro EL (rootfs) sin más que seleccionar una casilla de verificación. Algunas de las novedades son las siguientes:

#### OpenNTPD

Implementación del Network Time Protocol. <http://www.openntpd.org>.

---

linux, los device drivers, los módulos cargables y otras utilidades junto con las aplicaciones de usuario para ser todo ejecutado sobre sus módulos basados en ARM9.

 <b>CONTINEA</b> <small>Microprocesamiento modular + Conectividad</small>	<b>Embedded Linux 5.2</b>	Nota de Aplicación CoTC-007
	<b>Digi CC9C + Display VGA color</b>	Publicado: 00/00/0000 Página 3 de 17

### Python

Interprete de python y libraries para que puedan ejecutarse scripts de Python en el sistema. <http://www.python.org>.

### Net-SNMP

Conjunto de herramientas para implementación del Simple Network Management Protocol. Incluye daemons y cliente de línea de comandos.  
<http://net-snmp.sourceforge.net>.

### SQLite

Library en C que implementa un motor de base de datos SQL auto-contenido, "embebible", y "zero-configuration". Provee el gestor de BD sqlite3 y libraries que implementan la API sqlite, permitiendo crear aplicaciones basadas en esa ella.  
<http://www.sqlite.org>.

### ALSA utils

ALSA (Advanced Linux Sound Architecture). Es un set de drivers de hardware, libraries y utilidades que proveen funcionalidades de audio y MIDI.

### Ethtool

Ethtool es una herramienta de diagnóstico y optimización de red del kernel de Linux. <http://freshmeat.net/projects/ethtool/>.

## La instalación

Luego de instalar la distribución de Kubuntu en modalidad completa, tendremos un acceso directo en el escritorio para instalar el entorno de desarrollo "Digi ESP" basado en Eclipse para Linux embebido.


Por otra parte también se instalarán y configurarán automáticamente algunas herramientas tales como un servidor TFTP y NFS junto con sus directorios de operación y los permisos apropiados. Para completar la instalación debemos contar con una consola serial para interactuar con el módulo a través de su puerto serial A (placa de prototipos mediante) con esta conexión configurada en *38400,N,8,1* se podrá interactuar con el intérprete de comandos de U-Boot lo cual es indispensable para la puesta en marcha del sistema. Habiendo completado la instalación, se presentará un icono en el escritorio para el acceso directo al entorno de desarrollo y ya se podrá comenzar a trabajar con Linux embebido.

La primera vez, es muy recomendable ejecutar el Package Manager y aplicar todas las actualizaciones disponibles antes de comenzar a desarrollar.

Luego de la instalación hay que configurar el IDE de acuerdo con las características del hardware del módulo destino, desde el menú *Device Options -> Device Manager*:

- Click en el ícono etiquetado como "Digi EL" y luego en el ícono "New remote configuration"
- Completar el campo "IP Address" y presionar en "Validate Connection". Si la conexión falla se deben verificar los parámetros de la red.
- Pasar a la solapa "File Transfer" y elegir la opción adecuada, probablemente lo más simple sea FTP, click en "Apply" y la conexión será verificada
- En la última solapa ("Hardware") seleccionamos el procesador (NS9360 para nuestro caso) y luego el módulo que lo implementa (ConnectCore 9C para nuestro caso). Finalmente, elegimos la placa base para nuestras pruebas (CC-9C/CC-W9C dev. board) acá existe la opción de especificar una "Custom board" si es que el usuario definió su propia placa en el entorno de desarrollo<sup>5</sup>.

<sup>5</sup> Consultar el documento "Customizing Platform Code Digi Embedded Linux"

 <b>CONTINEA</b> <small>Microprocesamiento modular + Conectividad</small>	<b>Embedded Linux 5.2</b>	Nota de Aplicación CoTC-007
	<b>Digi CC9C + Display VGA color</b>	Publicado: 00/00/0000 Página 4 de 17

- Aplicar los cambios y establecer estas configuraciones como el hardware actual de trabajo con un click en "Set current".

Ahora podemos comunicarnos con el target, por ejemplo, desde el menú `Device Options -> Remote Target Inspector`. Allí es posible "leer" la lista de procesos actualmente corriendo en el módulo. Se desplegará una tabla de procesos y se pueden enviar mensajes al proceso seleccionado seleccionando la señal deseada desde el campo "Signal" correspondiente, todo esto sucede en realidad vía Telnet.

Otra característica interesante es que desde el menú `Device Options -> Update Flash` se pueden grabar las diversas particiones de la flash del módulo remotamente, sin necesidad de operar mediante la consola serial desde U-Boot.

Nota: quienes tengan en uso módulos con versiones anteriores de DEL tendrán que grabar una imagen generada con DEL5.2 tanto del U-Boot como del RootFS y por supuesto del Kernel para poder trabajar con módulos viejos.

## Los proyectos

El ciclo de configuración inicial, creación y configuraciones de un proyecto, debug y prueba de la aplicación de la versión actual no difiere significativamente de la versión antigua y dado que esto fue desarrollado en la nota CoAN-020, no vamos a repetir aquí el procedimiento.

Pero la nueva versión DEL5.2 incluye algunas particularidades relacionadas con los tipos de proyectos que se pueden desarrollar, como ser, la realización de proyectos de biblioteca (library) o de aplicaciones C++ basadas en interfaz gráfica de usuario utilizando el paquete de software Qtopia incorporado como plug-in en Eclipse.

Embedded Linux 4.0 ya incluía como ejemplo las librerías gráficas Qtopia y un par de aplicaciones de demostración que podían ser agregadas a priori a la imagen del sistema de archivos siempre que se hubiera incluido el componente "Root File System" en el proyecto. Estas librerías proporcionan un potente soporte gráfico para aplicaciones embebidas ofreciendo, soporte para mouse (el touch screen funcionará también como un mouse), manejo de ventanas, cajas de texto, manejo de eventos, etc. Pero esta vez, DEL5.2 va mucho más lejos, ofreciendo un entorno gráfico de desarrollo orientado a objetos, desde el cual es posible manipular e incluir en la aplicación del usuario componentes de interfaz tales como botones, cajas de texto, etc. y desarrollar el código de la aplicación en C++.


## Bibliotecas de funciones (library)

Una library, es un archivo que concentra funciones que son utilizadas repetidamente por varios programas. DEL5.2 incluye asistencia para la creación de libraries en C/C++ mientras que se puede optar por libraries estáticas o dinámicas (shared).

## Libraries estáticas vs. dinámicas (shared)

Una library estática contiene rutinas que son copiadas en una aplicación por el *linker*, produciendo un único programa autónomo. Básicamente son contenedores de recursos que son insertados en diferentes aplicaciones.

Por el otro lado, las funciones de las libraries dinámicas no son copiadas al ejecutable. El *linker* simplemente inserta en el programa algunos links a las libraries que necesita. Las libraries dinámicas residen como archivos separados en el sistema y su código es cargado en memoria cuando el programa se ejecuta. Estas libraries pueden ser compartidas por múltiples programas a la vez, por eso son llamadas también *shared libraries*. El uso de libraries estáticas produce ejecutables stand-alone, pero usualmente de gran tamaño. Las libraries dinámicas producen ejecutables pequeños, pero dependientes del sistema donde se ejecuten.

 <p>CONTINEA Microprocesamiento modular + Conectividad</p>	<h1>Embedded Linux 5.2</h1>	Nota de Aplicación CoTC-007
	<h2>Digi CC9C + Display VGA color</h2>	Publicado: 00/00/0000 Página 5 de 17

## Creación de un proyecto de biblioteca de funciones (Library)

Esta versión de DEL incluye un template llamado `Application/Library C/C++ Project` que facilita la tarea de escritura de proyectos de biblioteca. Para esto, hay que abrir el diálogo de creación de proyectos desde el menú `File->new->project...` y seleccionar:

- Digi EL Application/Library C Project    o    - Digi EL Application/Library C++ Project
- (fig. a)

A continuación se define el nombre del proyecto y la ruta de creación y luego se selecciona el tipo de library deseada:

- Static Library crea un proyecto C (o C++) del tipo static library.
  - Shared Library crea un proyecto C (o C++) del tipo shared library
- (fig. b)

Seguidamente, se seleccionan ambas configuraciones para el proyecto **Debug** y **Release**. fig. c

Y luego se configuran algunas características del archivo de biblioteca a generar, como el directorio de instalación y los permisos de usuario sobre el archivo. Fig. d

Al finalizar, el proyecto de biblioteca es creado y puede verse en el explorador de proyectos.

Para agregar nuevos archivos fuente a la library: `File->New->Source File`.

Para crear un nuevo header `File->New->Header File`.

Para generar el proyecto de biblioteca, utilizar los botones `Build`, `Rebuild` o `Install`.

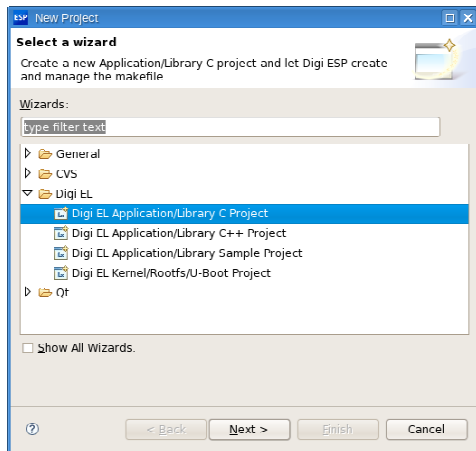


Fig. a

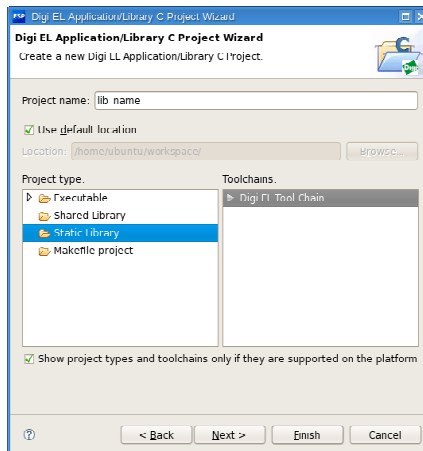


Fig. b

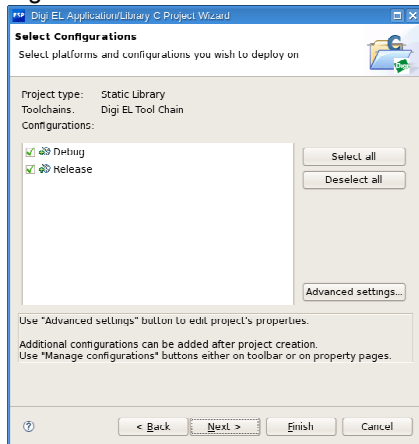


Fig. c

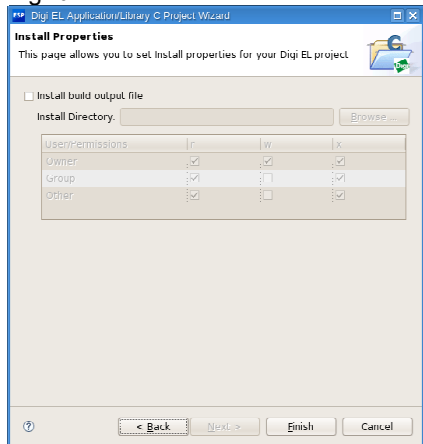



Fig. d

 Microprocesamiento modular + Conectividad	<b>Embedded Linux 5.2</b>	Nota de Aplicación CoTC-007
	<b>Digi CC9C + Display VGA color</b>	Publicado: 00/00/0000 Página 6 de 17

## Aplicaciones gráficas con QTDesigner

En esta versión se agrega una perspectiva de diseño para generar interfaces gráficas de usuario (GUI) en C++ mediante componentes Qtopia (**Trolltech's Qt Designer**). Esto permite emplear formularios y objetos gráficos en tiempo de desarrollo de manera tal que pueden previsualizarse tal como se verán luego en tiempo de ejecución.

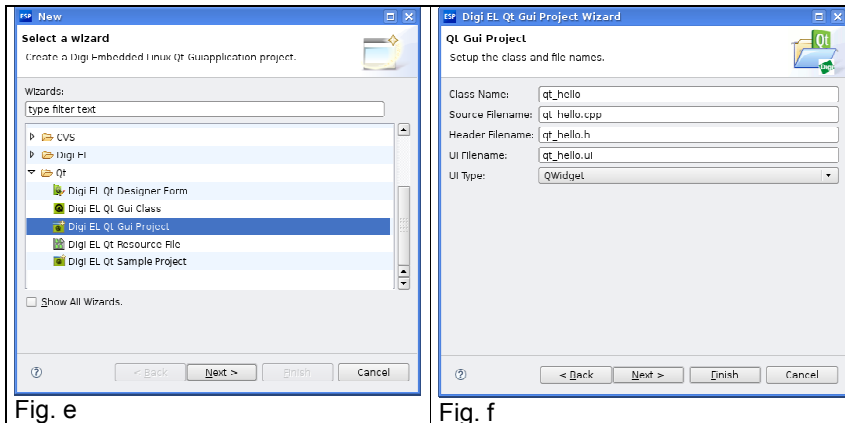
A continuación se verán los pasos para crear el clásico "Hello World" usando Qt Designer con el fin de que el lector se de una idea de las posibilidades que ofrece el entorno. Para profundizar referirse a la ayuda en línea:

[Help > Help Contents > Qt Development > Qt Development User Guide](#)  
 y  
[Help > Help Contents > Qt Development > Qt Reference Documentation](#)

### Creación del proyecto

Para crear el proyecto, nos dirigimos al menú **File -> New -> Project...** En el diálogo, expandimos la carpeta Qt y seleccionamos DigiEL Qt Gui project y click en Next. (Fig. e)

Nombramos el proyecto, por ejemplo como **qt\_hello**, presionamos **Next** y se abrirá un diálogo mostrando los archivos que se crearán utilizando el nombre suministrado por nosotros (Fig. f). En UI Type, también tenemos la posibilidad de seleccionar QDialog y QMainWindow.



Click **Next** y en la próxima ventana no seleccionar ningún módulo, solo click en **Finish** y un warning aparecerá para avisar que la perspectiva activa cambiará a la perspectiva Qt. Click **Yes**.

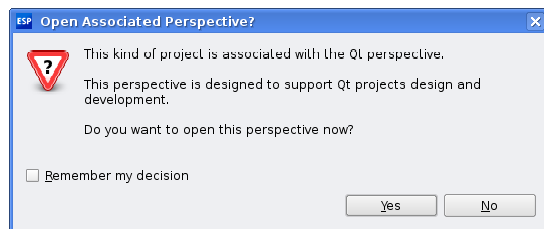


Fig. g

## La perspectiva Qt

La perspectiva Qt tiene la siguiente vista:

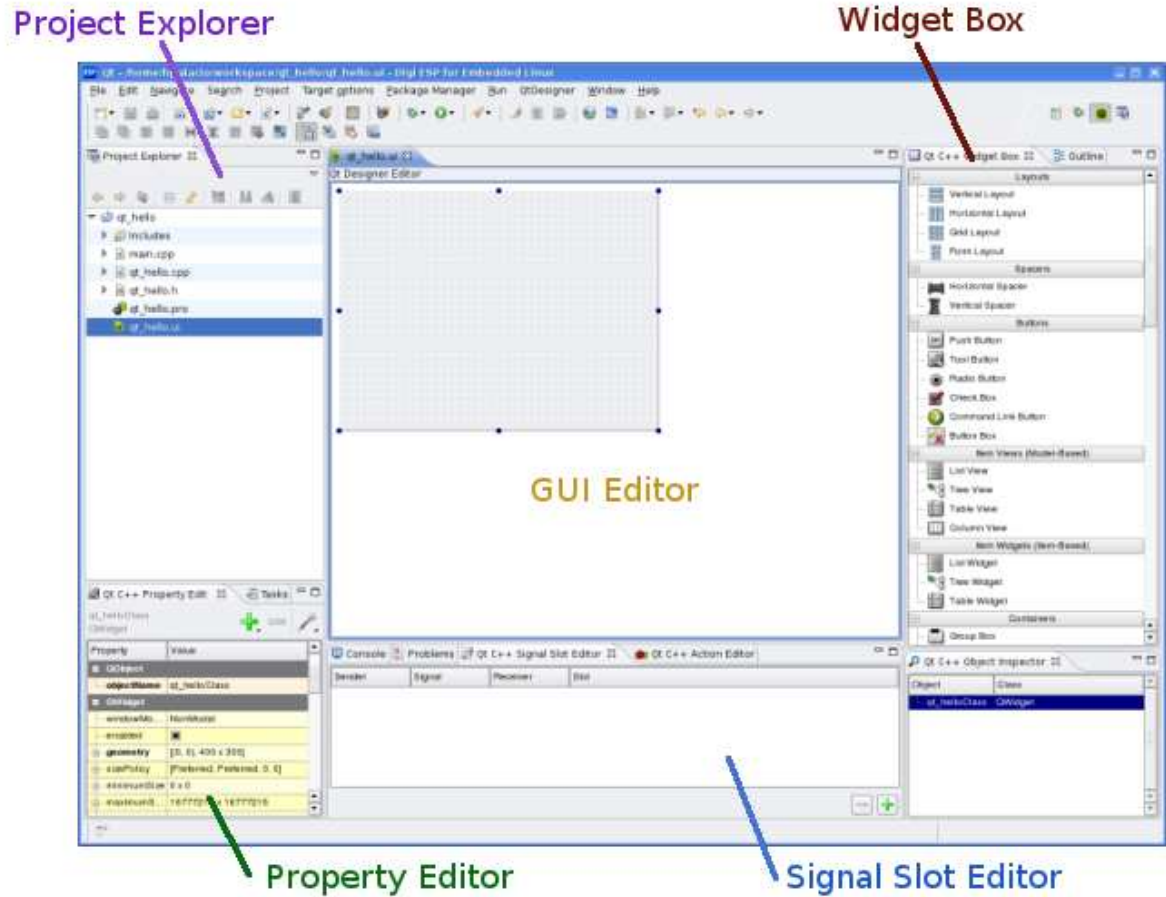


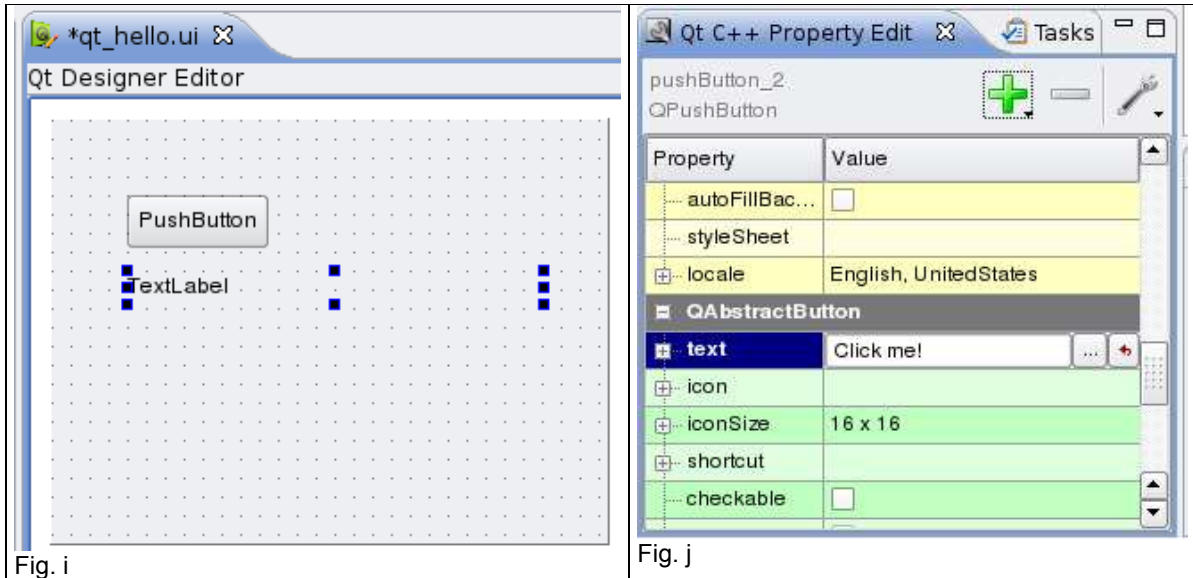
Fig. h

### Ubicar componentes en el formulario

En el "Project Explorer", abrir la interfaz gráfica de usuario haciendo doble click sobre `qt_hello.ui`. A continuación podremos ubicar algunos componentes allí mediante operaciones *drag and drop* desde el "Widget Box". Pondremos por ejemplo, un elemento *PushButton* en el formulario. Luego agregamos un *Label* de manera que quede con forma similar a la figura i.

Seleccionar el elemento "Button" en el formulario e ir al editor de propiedades. Localizar la propiedad *text* y cambiarla a "Click me!" o a lo que sea, bah, si no quiere, no la cambie.





Ahora, hacer lo mismo con el elemento *Label* del formulario, pero eliminando el texto de la propiedad *text*. El formulario quedaría así (Fig. k):

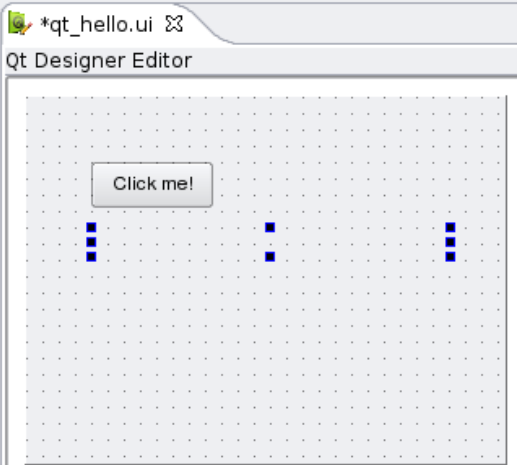


Fig. k

### Escribir código para la aplicación Qt

Nuestro ambicioso requerimiento es que el mensaje "Hello World" se muestre al usuario cuando presiona el botón. Para eso, necesitamos abrir el archivo `qt_hello.cpp` y agregar un método llamado **`on_x_clicked`** (donde *x* es el nombre de la instancia del botón en el formulario) donde nosotros establecimos la propiedad *text* del *Label*. El nombre del botón puede ser ubicado en el "Property Editor". Por default el botón es llamado 'pushButton' y la etiqueta es llamada 'label'.

Entonces, hay que agregar el código en **negrita** al archivo `qt_hello.cpp`:

```

#include "qt_hello.h"


qt_hello::qt_hello(QWidget *parent) : QWidget(parent)
{
    ui.setupUi(this);
}

qt_hello::~qt_hello()
{
}

void qt_hello::on_pushButton_clicked()
{
    ui.label->setText("Hello world!");
}

```



 <b>CONTINEA</b> <small>Microprocesamiento modular + Conectividad</small>	<b>Embedded Linux 5.2</b>	Nota de Aplicación CoTC-007
	<b>Digi CC9C + Display VGA color</b>	Publicado: 00/00/0000 Página 9 de 17

Ahora, abrir el header `qt_hello.h` y agregar la declaración del método a los slots públicos de la clase `qt_hello`:

```
#include <QtGui/QWidget>
#include "ui_hello_world.h"
class qt_hello : public QWidget
{
    Q_OBJECT

public:
    qt_hello(QWidget *parent = 0);
    ~qt_hello();

    public slots:
        void on_pushButton_clicked();

private:
    Ui::qt_helloClass ui;
};
```

## Compilar y correr la aplicación

Para compilar la aplicación ir al "Project Explorer", seleccionar el proyecto `qt_hello` y click en el botón Build, igual que cualquier otro proyecto. Para correr la aplicación en el target, antes se debe haber establecido los datos del módulo en "Target Remote Configuration".

Luego, create una "Run configuration"<sup>6</sup> y en el tab **Arguments**, incluir el argumento `-qws`, que lanza el X-Server en el sistema del target antes de correr la aplicación gráfica. Al ejecutar y presionar el botón se verá algo así:



Cabe destacar que para ejecutar correctamente las aplicaciones en el target, se deben cargar las aplicaciones QT de demostración en la imagen del rootFS, es decir que debemos preparar la imagen del sistema de archivos mediante el proyecto adecuado, compilarlo y actualizar la imagen del rootfs desde U-Boot.

## Conclusiones

Con la versión de Digi Embedded Linux 5.2 es posible utilizar eficientemente el entorno de desarrollo instalando la distribución de Linux contenida en el CD de instalación sobre máquinas virtuales.

Además, se contemplan una amplia gama de plataformas de hardware y con ello, una extensa disponibilidad de drivers de dispositivos que antes no estaban presentes.

Adicionalmente, contamos con la posibilidad de desarrollar aplicaciones con interfaces gráficas de usuario mediante herramientas gráficas de diseño orientadas a objetos, quizás esto último sea la novedad más significativa al comparar DEL4.0 con DEL5.2. En línea con todo esto, se agrega un nuevo tipo de proyecto para asistir al desarrollador en la creación de Libraries dinámicas o estáticas. En resumen, la versión de Digi Embedded Linux 5.2 presenta notables mejoras en cuanto a usabilidad y aceleración de tiempos de desarrollo, como así también en la evolución del IDE al agregar la perspectiva gráfica.

<sup>6</sup> Ver topic 2.12.1. *Launch Configuration* en la "Digi ESP™ for Embedded Linux – Documentation" disponible en el CD de instalación en formato HTML.