



UART Application Kit for Digi Embedded  
Linux  
User's Manual

© Digi International Inc. 2008. All Rights Reserved.

The Digi logo is a registered trademark of Digi International, Inc.

All other trademarks mentioned in this document are the property of their respective owners.

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

Digi International Inc.

11001 Bren Road East

Minnetonka, MN 55343 (USA)

\* +1 877 912-3444 or +1 952 912-3444

<http://www.digi.com>

# Contents

1	Introduction.....	6
<b>1.1</b>	<b>Features.....</b>	<b>6</b>
2	Requirements.....	7
<b>2.1</b>	<b>Host system requirements.....</b>	<b>7</b>
3	Installation.....	8
<b>3.1</b>	<b>Creating a new project.....</b>	<b>8</b>
4	Testing.....	9
<b>4.1</b>	<b>Load of driver.....</b>	<b>9</b>
<b>4.2</b>	<b>Simple test of ports.....</b>	<b>9</b>
<b>4.3</b>	<b>Using the serial ports.....</b>	<b>10</b>

# Conventions used in this manual

Here is a list of the typographical conventions used in this manual:

<b>Style</b>	New terms and variables in commands, code, and other input.
<code>style</code>	In examples, to show the contents of files, the output from commands. In, text the C code.
	Variables to be replaced with actual values are shown in italics.
<b>Style</b>	For menu items, dialogs, tabs, buttons, and other controls.
	In examples, to show the text that you enter
Menu name > option	A menu followed by one or more options; for example, File > New.

This manual also uses these frames and symbols:



---

**This is a warning. It helps solve or avoid common mistakes or problems.**

---



---

*This is a hint. It contains useful information about a topic.*

---



```
> This is a host computer session
> Bold text indicates what must be entered.
```



```
> This is a target session
> Bold text indicates what must be entered.
```

# Abbreviations

ASCII	American Standard Code for Information Interchange
CPU	Central Processing Unit
ESD	Electrostatic Discharge
FIFO	First In First Out
GPIO	General Purpose Input/Output
MEI	Multiple Electrical Interface
OS	Operating System
PC	Personal Computer
RAM	Random Access Memory
ROOTFS	Root File System
TFTP	Trivial File Transfer Protocol
TTL	Transistor Transistor Logic
UART	Universal Asynchronous Receiver Transmitter



# 1 Introduction

The UART Application Kit is a product that extends the number of serial ports available on your Digi Development board or JumpStart board.

The UART Application Kit board connects to the development board via the Peripheral Application Header and the provided 50-pin flat cable. It mounts a quadruple UART chip and additional logic for providing 4 independent configurable RS-232/422/485 serial ports.

Each port can be configured, via an on-board switch, for RS-232, RS-422 or RS-485 asynchronous serial communication in either full-duplex or half-duplex mode.

The UART Application Kit board is powered from the Development board.

The software package contains the driver needed to support the extended UARTs in Digi Embedded Linux.

This User's Manual assumes that the reader is able to create, compile and install a kernel + kernel modules project using Digi Embedded Linux environment.

## 1.1 Features

---

The following are the main features of the UART Application Kit:

- Four independent configurable RS-232/422/485 serial ports
- 64-byte transmit/receive FIFO with error flags
- Up to 5Mbits/sec baud rate
- Full modem RS-232
- Configurable full-duplex or half-duplex RS-422/485 ports
- Configurable termination resistor for RS-422/485 ports
- Four 9 way D type connectors
- All ports accessible at TTL level (connector not mounted)
- One shared open collector interrupt for all the ports
- Power save mode
- $\pm 15\text{kV}$  ESD protection



## 2 Requirements

### 2.1 Host system requirements

---

To create a DigiEL project with support for the UART Application kit, the Host PC should meet the following requirements:

- Linux OS
- Digi Embedded Linux 4.0 (or higher)



## 3 Installation

To install the software place the installer file **uart\_appkit\_installer.run** into any folder of a Linux host, and then execute it.



```
> ./uart_appkit_installer.run
This will install UART Application Kit software package
Do you want to proceed? [y]:
```

After you have accepted the license agreement you will be prompted for the installation folder of Digi Embedded Linux, and whether you want the installer to generate backups for the files it will replace.

The script installs the UART AppKit driver as a loadable module into your Digi Embedded Linux distribution. Any new project will include this module.

### 3.1 Creating a new project

---

The UART Application Kit driver has been designed as a loadable module based on the generic Linux 8250/16550 UART driver.

Create a new **kernel + kernel modules + rootfs** Digi EL project based on your hardware platform. The new project will include the driver for the UART Application Kit as a loadable module. You can find the source code in the folder **modules/uartkit/** of your project's root folder.

#### 3.1.1 Build the project

Compile and install the kernel and rootfs (with **make** and **make install**).

The UART AppKit driver will be stored as two loadable modules called **8250\_uak.ko** and **uart\_appkit.ko** within the project rootfs.





## 4 Testing

### 4.1 Load of driver

The UART Application Kit driver is based on a modified version of the generic 8250/16550 UART driver. The first thing to do is to load this driver:



```
> modprobe 8250_uak
Serial: 8250/16550 driver $Revision: 1.1.1.1 $ 4 ports, IRQ sharing enabled
```

Then, load the UART Application Kit driver:



```
> modprobe uart_appkit
uart_appkit V1.0
serial8250.0: ttySUAK0 at MMIO map 0x40000000 mem 0xc2864000 (irq = 29) is a
ST16654
serial8250.0: ttySUAK1 at MMIO map 0x40000008 mem 0xc2866008 (irq = 29) is a
ST16654
serial8250.0: ttySUAK2 at MMIO map 0x40000010 mem 0xc2868010 (irq = 29) is a
ST16654
serial8250.0: ttySUAK3 at MMIO map 0x40000018 mem 0xc286a018 (irq = 29) is a
ST16654
```

This will populate the target's file system with the following device nodes: **/dev/ttySUAK0**, **/dev/ttySUAK1**, **/dev/ttySUAK2**, and **/dev/ttySUAK3**.



**These modules are not designed to be unloaded from the kernel at runtime.**

### 4.2 Simple test of ports

To test the ports, connect a serial cable between two ports. For example port 0 and port 3.

Configure the ports in RS-232 mode (switch S1.1 is ON and rest are OFF).

Open a telnet session to the target and listen on port 0 with the *cat* command:



```
> cat /dev/ttySUAK0
```

Open a second telnet session and write to port 3 with the echo command (using redirection):



```
> echo "HELLO FROM PORT 3" > /dev/ttySUAK3
```

The message will be echoed into the session that listens to port 0.



## 4.3 Using the serial ports

---

The UART Application Kit serial ports can be handled like any other serial port in Linux. Please refer to the following documents for information about serial ports programming:

**Serial Programming HOWTO:**

<http://tldp.org/HOWTO/Serial-Programming-HOWTO/>

**Serial Programming Guide for POSIX Operating Systems:**

<http://www.easysw.com/~mike/serial/>