The High-Performance Alternative

Getting Started with PowerDNx Software

January 2010 Edition
Version 1.0
PN Man PowerDNxSW-0110

See UEI's website for complete terms and conditions of sale:

http://www.ueidaq.com/company/terms.aspx

Contacting United Electronic Industries

Mailing Address:

27 Renmar Avenue
Walpole, MA 02081
U.S.A.

For a list of our distributors and partners in the US and around the world, please see

http://www.ueidaq.com/partners/

Support:

Telephone:(508) 921-4600

Fax:(508) 668-2350

Also see the FAQs and online "Live Help" feature on our web site.

Internet Support:

Supportsupport@ueidaq.com
Web-Sitewww.ueidaq.com
FTP Siteftp://ftp.ueidaq.com

Product Disclaimers:

WARNING!

DO NOT USE PRODUCTS SOLD BY UNITED ELECTRONIC INDUSTRIES, INC. AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS.

Products sold by United Electronic Industries, Inc. are not authorized for use as critical components in life support devices or systems. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Any attempt to purchase any United Electronic Industries, Inc. product for that purpose is null and void and United Electronic Industries Inc. accepts no liability whatsoever in contract, tort, or otherwise whether or not resulting from our or our employees' negligence or failure to detect an improper purchase.

NOTE: Specifications in this document may change without notice. Check with UEI for current status.

Table of Contents

Chapter 1 Operating Modes

1.1 Overview

This chapter describes the various operating modes that you may employ in writing your application program. Chapter 2 describes basic characteristics of typical applications and the various APIs available to you for writing your program. Chapter 3 describes a procedure for choosing the operating mode and API best suited to your application.

1.2 Operating Modes

The term “operating mode” refers to the method of communication between a host and I/O module (IOM) in a typical data acquisition and control system. The IOM contains the hardware modules (layers) that connect to sensors, measuring equipment, and other real-world devices. In UEI systems, the communication link is an Ethernet link over a network.

When used with a standalone controller such as a UEIPAC (with no “host” involved), “operating mode” refers to the communication between the user space program and the I/O layer kernel driver.

The operating modes available for use with UEI’s PowerDNx Software are:

- **Immediate**

  In this mode, also called “Point-by-Point” or “Single Scan”, timing is controlled by the host. The host issues a request to the IOM for point by point input/output on each layer installed in the IOM. The IOM responds to each such request by sending the requested measurement data over the communication link. In other words, each request from the host requires one network request/response round trip. Because the host and IOM operate in lock-step, this is called synchronous operation. This mode may be used with all UEI layer types. Because of the lock-step nature of communication, this mode is quite inefficient and usually limited to a scan rate of less than 100 Hz, which is too slow for many applications.

  For example, an IOM with 12 AI layers requires 12 request packets to be sent from the host and 12 reply packets to be sent from the IOM with the data.

  If you are using DQEngine on a non-realtime OS, the DQEngine can report a lost packet to the user application. Since the immediate mode does not itself perform error correction except for serial communication layers such as SL-501, -504, and -508, the user application must handle any error correction provided.

  Immediate mode examples are named “SampleXXX”. All functions in these examples have the prefix DqAdv. . . ().

- **Buffered – ACB Advanced Control Buffer**

  In this mode, operation is started by a software or hardware trigger, timing is controlled by a hardware clock in the IOM, and it continues to run until stopped. This mode, as the name implies, sends data to circular buffers in the host or IOM (or other device) at maximum speed and verifies that each packet is completely filled with data before being sent. Since buffers are used and the host and IOM are not in lock-step, this is called asynchronous or buffered operation.

This mode is the most efficient operating mode for applications where it can be used.

A major feature of this mode is that it offers automatic error correction that prevents any loss of data in transmission. This feature, however, does present problems in meeting real time deadlines, which precludes its use in hard real time and stimuli/response control applications. Also, it is not supported for use with the UEIPAC.

ACB examples are named “SampleACBXXX”.

- **DMap (Data Mapping)**

  In this mode, timing is controlled by the host. When requested by the host, data from/to all layers in the IOM is input/output as a snapshot and all such data is transmitted to the host (or other device) in a single packet. In other words, each snapshot of all layers is transmitted in a single network round trip. (Note that if the data does not fit in a single packet, multiple packets can be used.) Note also that a user can combine different sets of channels into different DMaps and update them at different rates.

  This mode, therefore, is ideal for stimuli/response control and other realtime applications. DMap is the most efficient mode for point by point input/output on AI, AO, DIO, and CT layers. Since DMaps in host and IOM are continually matched and updated at all times, DMap is classified as a synchronous system.

  No error correction is available with this mode and there is no way to recover a lost packet. All packets are numbered, however, and DMap does notify the application if a packet is lost. Recovery of lost data is not possible. Although you are notified that data was lost, you have to move on, perhaps reusing the data from the last read. Note, however, that data loss on a dedicated network is extremely unlikely to occur (short of disconnecting the network cable or being in a very noisy EMI environment).

  DMap examples are named “SampleDMAPXXX”.

- **RtDMap (Real-time Data Mapping)**

  There are two DMap modes: DMap and RtDMap.

  The DMap mode uses the DqEngine and can asynchronously send events to the user program when the DMAP has been refreshed. It uses DqDmapXXX APIs. The refresh speed is limited to whatever timing service is provided by the OS. The maximum value is typically 1kHz under Windows and Linux.

  The RtDMap mode does not use the DqEngine. In this mode, It is the responsibility of the user application to refresh the DMAP periodically. It can therefore run faster than DMAP mode — as fast as the network will allow. It uses DqRtDmapXXX API.

  On a 100Mb/s network, the maximum refresh speed is about 4kHz, depending on the packet size (number of configured channels). On a 1Gb/s network, the maximum refresh rate is about 10kHz. On the UEIPAC, the maximum refresh rate is about 20 kHz.

  RtDmap examples are named “SampleRTDMAPXXX”.

- **Msg (Messaging)**

  This mode allows access to messaging layers (serial, CAN, ARINC, and MIL-1553) at full speed. Since it is designed to correct communication errors that may occur on the network, however, it is not suitable for real-time applications. Also, this mode is not supported on the UEIPAC. Because buffering is used, this mode is classified as asynchronous.

  Messaging mode uses an ACB to transfer messages. Messages can be grouped together in a single packet to improve performance. If desired, messages can be sent on receiving a timeout signal or upon receiving a specified amount of data.

  Messaging examples are named “SampleMSGXXX”.

- **VMap (Real-time Variable-size Data Mapping)**

  This mode, which is similar to DMap but with variable buffer size, allows access to all layers at full speed, transferring incoming and outgoing data in buffers in a single call. A single VMap can be used for the entire IOM (which may be partitioned into multiple VMaps as needed). A VMap packet can also be dynamically resized to optimize use of bandwidth. A VMap packet delivers output data and returns input data along with the number of samples written and the number still available.

  VMap examples are named “SampleVMapXXX”.

  VMap allows you to recover a lost packet. Since every VMap packet is sequentially numbered, you can request that an out-of-order packet be resent whenever one is received. VMaps hold data only for one previous packet.

- **VMap+ (Real-time Variable-size Data, Variable-channel Mapping)**

  This mode is similar to VMap with its variable buffer size, but also permits selection of channels on-the-fly. It is primarily designed for use with MIL-1553 messaging applications.

- **Other Modes – aDMap/aVMap, aEvent, Burst (not currently released)**

1.2.1 Operating Mode Support by Layer

Not all operating modes are supported by all layers. Before selecting your operating mode, be sure to verify that it supports your preferred operating mode. In case you forget to verify layer support, calling the 4 common functions **DqAcbIsSupported()**, **DqDmapIsSupporte()**, **DqVmapIsSupported()**, and **DqMsgIsSupported()** will automatically verify support for you. **Table 1-2** below lists the operating modes supported by each UEI layer type. Since new layers are frequently being added and software support extended, be sure to check with UEI for the latest updates regarding support for your layer types.

Table 1-1. Operating Mode Support by Layer Type (Analog Input)

Layer Type	Operating Modes Supported						
	Scan	**ACB**	**DMap**	**RtDmap**	**VMap***	**aEvent**	**Msg**
AI-201	Y	Y	Y	Y	Y	N	N
AI-202	Y	Y	Y	Y	Y	N	N
AI-205	Y	Y	Y	Y	Y	N	N
AI-207/208	Y	Y	Y	Y	Y	N	N
AI-211	Y	Y	Y	Y	Y	N	N
AI-217	Y	Y	Y	Y	Y	N	N
AI-224	Y	Y	Y	Y	Y	N	N
AI-225	Y	Y	Y	Y	Y	N	N
AI-254	Y	Y	Y	Y	Y	N	N
AI-255	Y	Y	Y	Y	Y	N	N

* UEIPAC support is implemented.
Hosted API support will be provided in the near future. Check with UEI for current status.

Table 1-2. Operating Mode Support by Layer Type (Analog Output)

Layer Type	Operating Modes Supported						
	Scan	**ACB**	**DMap**	**RtDmap**	**VMap***	**aEvent**	**Msg**
AO-302	Y	Y	Y	Y	Y	N	N
AO-308	Y	Y	Y	Y	Y	N	N
AO-308-350	Y	Y	Y	Y	Y	N	N
AO-308-353	Y	Y	Y	Y	Y	N	N
AO-308-420	Y	Y	Y	Y	Y	N	N
AO-328	Y	Y	Y	Y	Y	N	N
AO-332	Y	Y	Y	Y	Y	N	N
AO-333	Y	Y	Y	Y	Y	N	N
AO-358	Y	Y	Y	Y	Y	N	N

* UEIPAC support is implemented.
Hosted API support will be provided in the near future. Check with UEI for current status.

Table 1-3. Operating Mode Support by Layer Type (Digital Input/Output)

	Operating Modes Supported						
Layer Type	**Scan**	**ACB**	**DMap**	**RtDMap**	**VMap**	**aEvent**	**Msg**
DIO-401	Y	Y	Y	Y	N	N	N
DIO-402	Y	Y	Y	Y	N	N	N
DIO-403	Y	Y	Y	Y	N	N	N
DIO-404	Y	Y	Y	Y	N	N	N
DIO-405	Y	Y	Y	Y	N	N	N
DIO-406	Y	Y	Y	Y	N	N	N
DIO-452	Y						
DIO-416	Y	N	Y	Y	N	N	N
DIO-432/433	Y	N	Y	Y	N	N	N
DIO-448	Y	N	Y	Y	N	N	N
DIO-454	Y						

Table 1-4. Operating Mode Support by Layer Type (Messaging)

	Operating Modes Supported						
Layer Type	**Scan**	**ACB**	**DMap**	**RtDmap**	**VMap**	**aEvent**	**Msg**
1553-553	Y	N	N		Y	Y	N
CAN-503	Y	N	N		Y		Y
SL-501	Y	N	N		Y		Y
SL-508	Y	N	N		Y		Y
429-566	Y	N	N		Y		N
429-512	Y	N	N		Y		N

Table 1-5. Operating Mode Support by Layer Type (Counter/Timer or Quad Encoder)

	Operating Modes Supported						
Layer Type	**Scan**	**ACB**	**DMap**	**RtDmap**	**VMap**	**aEvent**	**Msg**
CT-601	Y	N	Y	Y	N	Y	Y
QUAD-604	Y	N	Y	Y	N		Y

1.2.2 Support of Operating Mode for UEIPAC

When used with a UEIPAC (with no "host computer" involved), the term "operating mode" refers to the communication between the user space program and the I/O layer kernel driver.

The UEIPAC SDK only supports Immediate, DMap, and VMap modes for its own "local" layers. ACB and Msg error-correcting modes can only be used for controlling layers remotely mounted in IOMs connected to the UEIPAC via Ethernet.

Table 1-6. Operating Mode Support for UEIPAC

<table>
<tr><th rowspan="2">Layer Type</th><th colspan="7">Operating Modes Supported</th></tr>
<tr><th>Immediate</th><th>ACB</th><th>DMap</th><th>RtDmap</th><th>VMap</th><th>aEvent</th><th>Msg</th></tr>
<tr><td>AI-201</td><td>Y</td><td rowspan="29">Y
for remote layers only

N
for local layers</td><td rowspan="29">Y
for remote layers only

N
for local layers</td><td rowspan="29">Y
for all non-msg layers (local and remote)</td><td>Y</td><td>N</td><td rowspan="29">Y
for remote layers only

N

for local layers</td></tr>
<tr><td>AI-205</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AI-207</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AI-208</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AI-211</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AI-225</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AI-254</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AI-255</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AO-302</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AO-308</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>AO-332</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>DIO-401</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>DIO-402</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>DIO-403</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>DIO-404</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>DIO-405</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>DIO-406</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>DIO-416</td><td>Y</td><td>N</td><td>N</td></tr>
<tr><td>DIO-432</td><td>Y</td><td>N</td><td>N</td></tr>
<tr><td>DIO-433</td><td>Y</td><td>N</td><td>N</td></tr>
<tr><td>DIO-448</td><td>Y</td><td>N</td><td>N</td></tr>
<tr><td>CT-601</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>QUAD-604</td><td>Y</td><td>Y</td><td>N</td></tr>
<tr><td>SL-501</td><td>Y</td><td></td><td>N</td></tr>
<tr><td>SL-508</td><td>Y</td><td></td><td>N</td></tr>
<tr><td>CAN-503</td><td>Y</td><td></td><td>N</td></tr>
<tr><td>429-566</td><td>Y</td><td></td><td>N</td></tr>
<tr><td>429-512</td><td>Y</td><td></td><td>N</td></tr>
<tr><td>1553-553</td><td>Y</td><td></td><td>N</td></tr>
</table>

1.2.3 Support of Operating Mode by OS

Not all modes are supported for all operating systems. The major difference is that buffered modes and those that offer error correction cannot be used with realtime systems or the UEIPAC.

Table 1-7. Operating Mode Support by Operating System

Operating System	Operating Modes Supported						
	Scan	ACB	DMap	RtDmap	VMap	aEvent	Msg
Windows / Linux	Y	Y	Y	Y	Y	Y	Y
Realtime	Y	N	N	Y	Y	N	N
UEIPAC	Y	N	Y	N	Y	N	Y
QNX	Y	Y	Y	N	Y	N	Y

1.2.4 Support of Operating Modes by Environment

Not all modes are supported for all environments. The major difference is that buffered modes and those that offer error correction cannot be used with realtime systems or the UEIPAC. Also, there is no Framework support as yet for VMap or 3rd party Application Drivers.

Table 1-8. Operating Mode Support by Environment

Operating System	Operating Modes Supported						
	Scan	ACB	DMap	RtDmap	VMap	aEvent	Msg
Low-level C API (PDNALib)	Y	Y	Y	Y	Y	Y	Y
Framework	Y	Y	Y	N	N	N	Y
3rd part App Drivers	Y	Y	Y	N	N	N	Y
Realtime Library	Y	Y	N	Y	Y	N	N

1.3 Key Attributes of Modes with Error Correction

The key attributes of error-corrected, lossless operating modes are as follows:

- **ACB** — continuous stream of data from AIn/DIn or to AOut/DOut
- **Msg** — messages from/to Serial, CAN, ARINC, or 1553 layers
- **Timing Control** — All timing is defined by real-time clock inside IOM

1.4 ACB Mode Features

Key features of the ACB Mode are:

- Each subsystem of each layer (Analog Input, Analog Output, Digital Input/Output, etc.) is handled as a separate data stream.
- Since delivery of every data point is guaranteed, data from various layers is synchronized by definition.
- ACB data can be synchronized with other data by use of timestamps and the SyncX interface
- Use of ACB allows devices to be clocked externally.
- Stream-to-memory sub-mode (Burst Mode[1])
  --Improves performance by storing data first into RAM and then transferring it on a stop trigger.
  --64MB of RAM is available for temporary data storage (equal to four seconds of data from four AI-205 layers)
  --Cannot work continuously
- Can stream data on change of state of digital input sub-mode
- Can be used in raw data or converted data mode

1.5 Messaging Mode Features

Key features of Messaging mode are:

- Uses ACB to transfer messages and guarantee their delivery
- Allows messages to be grouped together in single packet to improve performance.
- Allows messages to be sent upon receiving a specified amount of data or upon expiration of a specified time period.

1.6 Immediate Mode Features

Key features of Immediate modes are:

- DMap — direct data mapping between host and IOM, fixed data size
- VMap — direct data mapping between host and IOM, variable data size
- Two operating methods:
  --Using DQEngine with non-realtime OS, DQEngine reports lost packet and guarantees message continuity
  --Using realtime functions with realtime or non-realtime OS, user application handles all error correction.

1. Burst Mode has been removed from 5200 and GigE CPUs.

1.7 DMap Mode Features

Key features of Dmap mode(s) are:

- Single DMap for whole IOM (all layers/subsystems) can be used
- DMap packet delivers output data, IOM returns most recent input data as reply
- Reply is guaranteed in 250us (133us for GigE system)
- All layers in IOM synchronized by definition
- Data can be synchronized across multiple IOMs based on host requests (guarantees that data is synchronized within DMap timebase)
- DMap notifies user application if packet is lost (no recovery is available)

1.8 VMap Mode Features

Key features of VMap mode are:

- Single VMap for whole IOM (may be partitioned into multiple VMaps)
- VMap packet delivers output data and returns input data along with number of samples written and number available
- VMap packet can be dynamically resized to optimize use of bandwidth
- VMap has built-in mechanism to notify user application about lost packet (no recovery of a lost packet is available)

1.9 VMap+ Mode Features

Key features of VMap+ mode are basically the same as for VMap with its variable buffer size, but this mode also permits selection of channels on-the-fly. It is primarily designed for use with MIL-1553 messaging applications.

Chapter 2 Characteristics of Typical Applications

2.1 Typical User Application Description

The basic characteristics of an application that are important to consider in selecting the right operating mode for your program are:

- Need to deliver all data without any gaps — no lost data.
- Need to deliver data on time with no delays and no missed deadlines
- Ability to tolerate some delays in delivery — up to 100 ms
- Ability to tolerate some missing data

The three general types of systems in which UEI equipment is commonly used and their typical characteristics are:

- **Data Acquisition**
  Need to deliver all data without gaps. Minor delays (1-2 seconds), however, can usually be tolerated.
- **Control**
  Need to deliver data on time with only short (0.5-10 ms) delays acceptable. A few missing points can usually be tolerated. (Timeliness is more important than gaps in data.)
- **Complex Application**
  Deliver all data in shortest time possible. A 10 ms delay in delivery of control loop data is usually acceptable.

2.1.1 Operating System Capabilities

Another characteristic of a user application that is important to consider in selecting the operating mode for your application is the Operating System used. The Operating Systems typically used with Data Acquisition/Control Systems fall into two broad categories:

- **Non-realtime (soft realtime) Operating Systems,** such as
  --Windows XP: 10ms soft-realtime
  --Linux 2.6: 10 ms soft-realtime, (1 ms with pre-emptive patch)
- **Real-time Operating Systems**
  --Realtime Linux, Windows RTX: 250us hard-realtime control loops

Note that all operating modes are supported for use with the non-realtime operating systems. Only DMap, VMap, and Single Scan Immediate are supported with realtime operating systems, however. No buffered modes such as ACB or Msg can be used.

2.1.2 Attributes of a Typical ACB/MSG Application

The characteristics of a typical application that would be a good candidate for using an ACB or Messaging operating mode are:

- Written for non-realtime Windows or Linux OS, uses LabVIEW or DASYLab as a frontend.
- Collects all data from AIn/DIn sensors at 10kS/s.
- Sends outputs to AOut/DOut at 1 kHz.
- Generates waveforms/patterns at 1-100 kHz.
- Sends requests to serial and CAN buses at 100fps.

- Collects replies from serial and CAN buses at 100fps.
- Invokes main application loop 10-20 times/second to read and process input and prepare output.

2.1.3 Attributes of a Typical DMap/VMap Application

The characteristics of a typical application that would be a good candidate for using a DMap or VMap operating mode are:

- Written for realtime or non-realtime OS.
- Host defines timing, requires immediate delivery of most recent data
- Loop time data rates from 50-500Hz to 100Hz-4kHz (RT).
- Invokes main loop upon receiving a new slice of data, then processes it, and updates outputs.
- Messages are sent/received with rates that cannot overflow FIFOs within selected control loop time.
- Diagnostics are required (using the secondary Ethernet port).

2.2 APIs Available

The various APIs available to you for use in writing the program for your application are:

- **Framework API** —

  UEI's Framework is a high-level API that is designed to be easy to use in most applications and with a wide variety of development environments. It is object-oriented and has bindings for use with Visual Basic, Excel, C/C++, VB.NET, C#, ActiveX, OPC, DASYLab, LabVIEW, MATLAB, OPC, and others. Several sets of example Framework programs are available to a user. Where possible, UEI recommends that users employ the Framework API in writing their application programs. The Framework API is fully described in the Framework Reference Manual and the Framework User Manual, both available for download at www.ueidaq.com.

- **PowerDNx API** —

  The PowerDNx API is a complete set of low-level (and some high-level) functions and software tools designed specifically for managing UEI hardware layers. The PowerDNx API is an extremely thin layer designed for maximum performance. To save CPU cycles, it does not validate all supplied parameters. These tools provide "hands-on" control for every layer in the UEI product line. They are fully described in the PowerDNx API Reference Manual, which is available for download from www.ueidaq.com. An extensive set of program examples is available to the user.

- **ACB API —**

  The ACB API is a subset of functions within the PowerDNx API that are designed to work with systems running in the ACB operating mode. They are described in the PowerDNx API Reference Manual. An extensive set of program examples (called SampleAcbXXX) is available to the user.

- **DMap API** —

  The DMap API is a subset of functions within the PowerDNx API that are designed to work with systems running in the DMap operating mode. They are described in the PowerDNx API Reference Manual. An extensive set of program examples (called SampleDMAPXXX) is available to the user.

- **RtDMap API** —

  The RtDMap API is a subset of functions within the PowerDNx API that are designed to work with real-time systems running in the RtDMap operating mode. They are described in the PowerDNx API Reference Manual. A set of program examples (called SampleRTDMAPXXX) is available to the user.

- **VMap API** —

  The VMap API is a subset of functions within the PowerDNx API that are designed to work with systems running in the VMap operating mode. They are described in the PowerDNx API Reference Manual. An extensive set of program examples (called SampleVmapXXX) is available to the user.

- **VMap+ API** —

  The VMap+ API is a subset of functions within the PowerDNx API that are designed to work with systems running in the VMap+ operating mode. They are described in the PowerDNx API Reference Manual. A set of program examples is available to the user (called SampleVMap+XXX).

- **Event API —**

  The ACB and Dmap Control and Event API is a subset of functions within the PowerDNx API that are designed to work with systems running in the ACB and DMap operating mode that are controlled by outside events. They are described in the PowerDNx API Reference Manual.

Chapter 3 Choosing Your Operating Mode

3.1 Choosing the Best Operating Mode

Use the following procedure to select the best operating mode for your application:

STEP 1: Determine the specific needs of your application.

1. What is the most important attribute for your application – data integrity, scanning speed, or timely delivery of data?
2. Can you tolerate delays in data delivery? If so, how long?
3. Can you tolerate gaps in your data? How many, how often, and how long?
4. Are you using Messaging? What type(s)?
5. What type of application are you designing — data acquisition only, closed-loop control system, stimulus/response system, or a complex combination system?
6. What types of measurement subsystems do you have? AI, AO, DI, DO, Msg, Other?
7. What layers are you using? Are they supported by ACB, DMap, VMap?
8. What sampling rate(s) do you need for the application? Below 100Hz? Above 100 Hz? 1000 Hz? Other?
9. What operating system are you using? Non-realtime? Realtime?
10. If using a realtime OS, are you planning to use ACB or Messaging? (Not supported)
11. Are you using Realtime library?
12. Does collected data need to be timestamped and synchronized? Across subsystems? Across entire IOM?
13. Are you using QNX OS?

STEP 2: Choose your operating mode(s).

STEP 3: Verify that your operating mode is supported by each specific layer and by your Operating System.

STEP 4: Implement your choice by (1) selecting applicable code samples for each layer, (2) adapting the samples to your specific needs, and/or (3) writing your own code as needed to meet the specific requirements of your application.

3.1.1 Features of Immediate Mode

Typical characteristics of the Immediate operating mode are:

- Simplest operating mode to use, supports all UEI layers
- Least efficient, slowest mode, limited to about 100 Hz scan rate
- May be selected on a per layer basis

3.1.2 Features of ACB Mode

Typical characteristics of the ACB operating mode are:

- Not hard realtime
- Acquires/stores/displays data
- Data is collected continuously at more than 10Hz and is transmitted with no gaps or errors because of buffers and automatic error correction

- Delivery delays are typically within the range of 0.1 to 1 second
- Timing is controlled by hardware clock in IOM, which can be triggered from an external source or event.
- Data stream is greater than 10kB/s
- May be selected on a per layer basis

3.1.3 Features of DMap Mode

Typical characteristics of the DMap operating mode are:

- Well-suited for control and/or simulation application
- Host controls timing, needs minimum response time
- Must receive current input data and write output data with zero delay
- Does not tolerate network collisions
- With non-realtime OS, requires scan rate of 100 to 500Hz. (up to 4 kHz with realtime OS)
- System may have multiple IOMs
- May be selected on a per layer basis

3.1.4 Features of VMap Mode

Typical characteristics of the VMap operating mode are:

- Suited for control and/or simulation application
- May have realtime data of a size larger than one scan or may have variable length messages such as CAN or Serial
- Main operating mode used in application is DMap
- Host controls timing
- Needs maximum performance and bandwidth of IOM with minimum response time
- May need advanced features such as message scheduler, frame delays, and data repetitions
- May be selected on a per layer basis

3.1.5 Features of Messaging Mode

Typical characteristics of the Messaging operating mode are:

- Used to send, receive, store a stream of messages
- Can tolerate minor delays in delivery
- Guaranteed message delivery, no missing data
- Maximum communication bus loads (serial, CAN, ARINC, 1553)
- Receive data up to specified no. of bytes, messages, defined content, or period of time
- Uses buffers for error correction like ACB
- May be selected on a per layer basis

3.1.6 Features of ACB/Msg Mode

Typical characteristics of the ACB/Msg operating mode are:

- May need either non-realtime Windows or Linux, with LabVIEW or DASYLab as a front-end
- Can tolerate minor delays in delivery
- Collects all data from AIn/DIn sensors at up to 10kS/s
- Sends output to AOut/DOut at 1 kHz
- Can generate waveforms/patterns at 1-100kHz
- Sends requests to serial and CAN buses at 100 frames/s
- Collects replies form serial and CAN buses at 100 frames/s
- Invokes main application loop 10-20 times per second to read and process input and prepare output
- Uses buffers for error correction like ACB
- May be selected on a per layer basis

3.2 Examples

This section is a group of examples of typical sets of layers and IOMs for use in specific applications. The purpose of the examples is to illustrate the procedure you should use to choose the best operating mode(s) for writing your application program.

Application Description:

- Type of IOM (PPCx Cube, GigEx Cube, DNR-6-1G, DNR-12-1G, UEIPAC)
- Layers Used
- No/type of inputs/outputs each layer
- Operating System: non-RT / RT
- Required Scan Rate: < 100 Hz | 100-500 Hz | 500-1000 Hz, | >1kHz
- Simultaneous Sampling Required? Y/N
- Can the application tolerate missed data points, say once every 10^6-10^7 samples?
- Delay Tolerated: < 250us / 0.5-10ms / 10-20 ms / 1-2 sec
- Stimulus/response? Y/N
- Control system? Y/N

3.2.7 Example 1

This is an example of a non-RT typical data acquisition system application.

3.2.7.1 Description

6-layer PPC8-1G Cube

Layer 0 -- AI-207, scan rate of 1 kHz, 16 differential inputs, T/Cs
Layer 1 -- AI-211, scan rate of 2.5 kHz, 4 IEPE inputs, 0-8 mA excitation output
Layer 2 -- DIO-403, scan rate 1.5 kHz, 8 DI, 8 DO
Layers 3,4,5 -- not used

Windows XP, non-RT, minor delays tolerated, not control, not stim/response, 100% data integrity

3.2.7.2 Mode Select

Table 3-1. Example 1 — Mode Selection Matrix – PPC8-1G 6-layer

Layer No.	Layer Type	Sample Rate	Possible Operating Modes					
			Immediate	**ACB**	**DMap**	**RtDmap**	**VMap**	**Msg**
0	AI-207	1Khz	N	Y	Y	N	Y	--
1	AI-211	2.5 kHz	N	Y	Y	N	Y	--
2	DIO-403	1.5 kHz	N	Y	Y	N	Y	--
3, 4, 5	--	--	--	--	--	--	--	--
See Note:			**1**	**2**	**3**	**4**	**5**	**--**

Note 1 – Scan rates too high for Immediate mode.
Note 2 – ACB mode is acceptable for all layers.
Note 3 – DMap is acceptable for all layers.
Note 4 – RtDMap not applicable -- not a realtime app.
Note 5 – VMap is acceptable for all layers.

Conclusion: Recommend ACB for all layers because it is most efficient.

3.2.8 Example 2

This is an example of a real-time simulator application.

NOTE: This example is valid when VMap becomes available for all digital and analog layers (expected soon – check with UEI for status)

3.2.8.1 Description

12-slot DNR-12-1G RACKtangle chassis

Slot 0 -- AI-207, scan rate of 1 kHz, 16 differential inputs, T/Cs
Slot 1 -- AI-211, scan rate of 100 kHz,
Slot 2 -- DIO-403, scan rate 1 kHz, 8 DI, 8 DO

Layers 3,4,5 -- not used

Windows XP, non-RT, minor delays tolerated, not control, not stim/response, 100% data integrity

3.2.8.2 Mode Selection

Since this is a hard realtime simulator application, ACB buffered modes cannot be used except for the ARINC 429 messaging layers. Therefore, DMap or RtDmap operating mode is recommended.

Therefore, the best choice of operating mode for this application is DMap, or RtDmap for the sensors and VMap+ for ARINC layers.) (UEI recommends using similar APIs wherever possible.)

Table 3-2. Example 2 — Mode Selection Matrix – DNR-12-1G -- RTX OS

Layer No.	Layer Type	Sample Rate	Possible Operating Modes					
			Immediate	ACB	Msg	DMap	RtDMap	VMap
0	AI-207	1Khz	N	N	--	Y	Y	Y*
1	AI-211	100kHz	N	N	--	Y	Y	Y*
2	DIO-433	1 kHz	N	N	--	Y	Y	Y*
3	DIO-433	1 kHz	N	N	--	Y	Y	Y*
4	DIO-448	1 kHz	N	N	--	Y	Y	Y*
5	DIO-448	1 kHz	N	N	--	Y	Y	Y*
6	AO-332	10 kHz	N	N	--	Y	Y	Y*
7	AI-255	1 kHz	N	N	--	N	N	Y
8	AI-255	1 kHz	N	N	--	N	N	Y
9	429-566	1 MB/s	N		Y	N	N	Y
10	429-566	1 MB/s	N	--	Y	N	N	Y
11	--							
See Note:			**1**	**2**	**--**	**3**		**4**

*Not yet available. Check with UEI for current status.

Note 1 – Scan rate too high for Immediate mode.
Note 2 – ACB mode is NOT acceptable for hard realtime.
Note 3 – DMap is acceptable for all layers except 429-566.
Note 4 – VMap is acceptable for all layers.

Conclusion: Use VMap for all analog and digital layers.

3.2.9 Example 3

This is an example of a non-RT data acquisition system

3.2.9.1 Description

6-layer DNA-PPC8-1G Cube chassis

Layer 0 -- DIO-406 12 inputs/12 outputs (1A current sinks), 5 kHz
Layer 1 -- AI-207, 16 differential inputs (T/Cs), scan rate of 1 kHz
Layer 2 -- AI-211, 4 inputs, scan rate of 2.5 kHz
Layer 3 -- DIO-403, 8 DI, 8 DO, scan rate 1.5 kHz,

Layers 4 and 5 -- not used

Windows XP, non-RT, minor delays tolerated, not control, not stim/response, 100% data integrity

3.2.9.2 Mode Selection

Scan rates of all layers are too high for use of immediate mode for any layer. Since the OS is non-realtime and delays can be tolerated, ACB can be used efficiently. DMap may also be used, if preferred.

Therefore, the best choice of operating mode for this application is ACB.

Table 3-3. Example 3 — Mode Selection Matrix – PPC8-1G 6-layer

Layer No.	Layer Type	Sample Rate	Possible Operating Modes					
			Immediate	**ACB**	**Msg**	**DMap**	**RtDmap**	**VMap**
0	DIO-406	5kHz	N	Y	--	Y	N	Y
1	AI-207	1kHz	N	Y	--	Y	N	Y
2	AI-211	2.5 kHz	N	Y	--	Y	N	Y
3	DIO-403	1.5 kHz	N	Y	--	Y	N	Y
4	--	--	--	--	--	--		--
5	--	--	--	--	--	--		--
See Note:			**1**	**2**	**--**	**3**	**4**	**5**

Note 1 – Scan rate too high for Immediate mode.
Note 2 – ACB mode is acceptable for all layers.
Note 3 – DMap mode is acceptable for all layers.
Note 4 – RtDmap not used because system is not realtime.
Note 5 – VMap mode is acceptable for all layers.

Conclusion: Use ACB for all layers (most efficient).

3.2.10 Example 4

This is an example of a 6-layer PPC8-1G non-RT data acquisition system with some CAN messaging.

3.2.10.1 Description

6-layer DNA-PPC8-1G Cube chassis

Layer 0 -- DIO-406 12 inputs/12 outputs (1A current sinks), 5 kHz
Layer 1 -- AI-207, 16 differential inputs (T/Cs), scan rate of 1 kHz
Layer 2 -- CT-601, 4 inputs, scan rate of 2.5 kHz
Layer 3 --CAN-503
Layer 4 - AO-308, 8 analog voltage outputs, scan rate of 50 kHz

Layers 5 -- not used

Windows XP, non-RT, minor delays tolerated, not control, not stim/response, 100% data integrity, CAN messaging used

3.2.10.2 Mode Selection

Scan rates of all layers preclude use of immediate mode for any layer. VMap and DMap are acceptable for all layers except the 503. Since the OS is non-realtime and delays can be tolerated. ACB can be used efficiently for all layers except CAN-503, which should use Msg mode.

The use of ACB is recommended for all layers in this application except CAN-503, which should use Msg mode.

Table 3-4. Example 4 — Mode Selection Matrix – PPC8-1G 6-layer

Layer No.	Layer Type	Possible Operating Modes					
		Sample Rate	Immediate	ACB	Msg	DMap	VMap
0	DIO-406	100Khz	N	Y	--	Y	Y
1	AI-207	1kHz	N	Y	--	Y	Y
2	CT-601	1 kHz	N	Y	--	Y	Y
3	CAN-503	1 kHz	N	--	Y	N	N
4	AO-308	50 kHz	N	Y	--	Y	Y
5	--	--	--	--	--	--	--
See Note:			**1**	**2**	**2**	**3**	**4**

Note 1 – Scan rate too high for Immediate mode.
Note 2 – ACB mode is acceptable for all layers except 503.
Note 3 – DMap mode is acceptable for all layers except 503.
Note 4 – VMap mode is acceptable for all layers except 503.

Conclusion: Use ACB for all layers except 503. Use Msg for 503.