

Акционерное общество
«Научно-технический центр Единой энергетической системы»
АО «НТЦ ЕЭС»

Программа с сетевым интерфейсом –
менеджер ресурса 32-канальной платы ЦАП
PCI 1724U (DCOM PCI Server)

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

	2
ВВЕДЕНИЕ.....	3
1. РАБОТА С ПРОГРАММОЙ.....	3

ВВЕДЕНИЕ

В руководстве приведена краткая инструкция по работе с программой с сетевым интерфейсом – менеджер ресурса 32-канальной платы ЦАП *PCI 1724U (DCOM PCI Server)*.

1. РАБОТА С ПРОГРАММОЙ

После установки и настройки программы дальнейший контроль ее работы осуществляет операционная система *Windows* посредством использования модели *DCOM*.

Написанные с помощью поставляемых вместе с программой компонентов для разработки программные средства должны подключаться в соответствии с описанным в настоящем документе и Описании программы алгоритмом.

Внешний вид программы *DCOM PCI Server* представлен ниже.

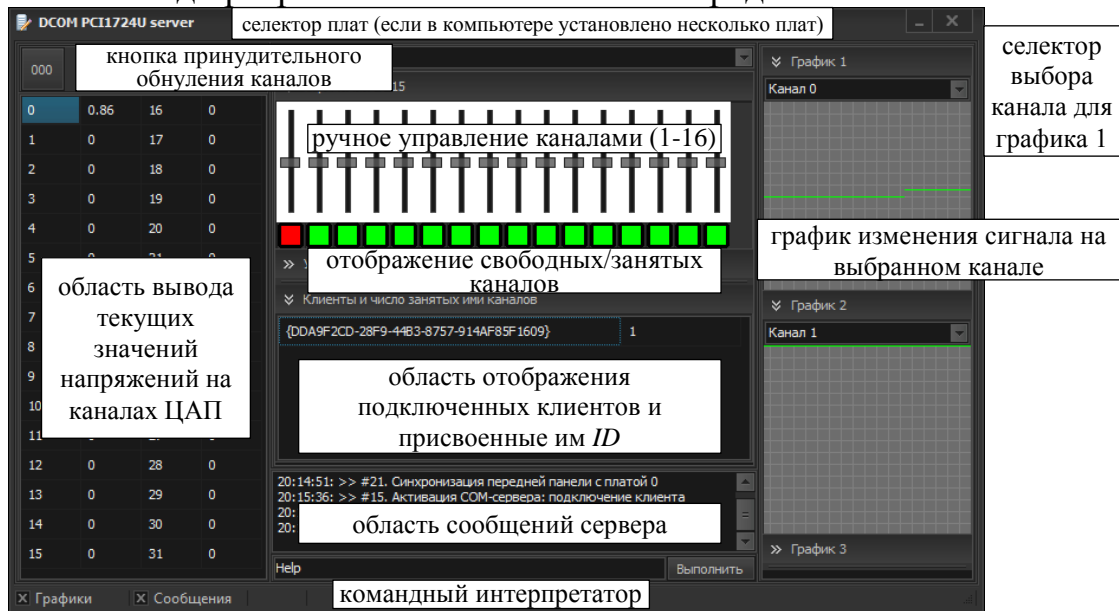


Рисунок 1 – Экранная форма программы *DCOM PCI1724U*

Программа содержит сервер и встроенный клиент, посредством последнего пользователь может с экранной формы управлять свободными каналами платы.

Интерфейс программы масштабируется двойным щелчком мыши по отдельным элементам.

В левой части программы *DCOM PCI Server* располагается таблица, в которой выводятся текущие значения напряжений на каждом канале ЦАП (нумерация 0-31).

Выше таблицы располагается кнопка «000», принудительно устанавливающая напряжения на всех занятых серверным клиентом каналах ЦАП напряжением уровнем 0 В. Для работы кнопки для одного или

нескольких каналов ЦАП, они должны быть захвачены клиентом, работающим в программе путем изменения напряжения с помощью слайдеров (см. ниже).

В средней части располагаются панель, содержащая 32 слайдера (по одному для каждого канала ЦАП). С помощью слайдеров пользователь может установить на любом незанятом другими клиентами канале ЦАП напряжение (± 10 В). Занятые другими клиентами каналы отображаются красной лампой. Канал может быть принудительно освобожден пользователем со стороны сервера, если щелкнуть мышью по красной лампе и согласиться с предложением освободить канал принудительно.

По умолчанию вся индикация на экранной форме выключена (отсутствует галочка «Сообщения»). Установка галочки включает всю индикацию состояния сервера на экранной форме.

Ниже по центру располагается таблица, в которой перечисляются идентификационные номера клиентов (*GUID*) и количество каналов, занятых каждым клиентом.

Ниже по центру располагается область уведомлений, в которой для пользователя сервера отображается текстовая информация о событиях сервера (подключение клиентов, выдача каналов, ошибки на стороне сервера/клиента и т.д.).

Ниже по центру располагается командный интерпретатор (реализовано выполнение трех команд *help*, *howuse*, *clear*).

В правой части окна располагаются три графика-индикатора, отображающие уровни и характер изменения напряжений по каждому из трех выбранных каналов.

Программа использует *COM*-функции, описанные в файле «C:\Advantech\DAQNavi\Inc\bdaqctrl.h» (файлы устанавливаются на компьютер при установке программы *Advantech Navigator*).

Программа содержит описание библиотеки типов (**.ridl*), в котором объявлены следующие функции (описание реализуемых в ходе вызова функций проверок не приводится):

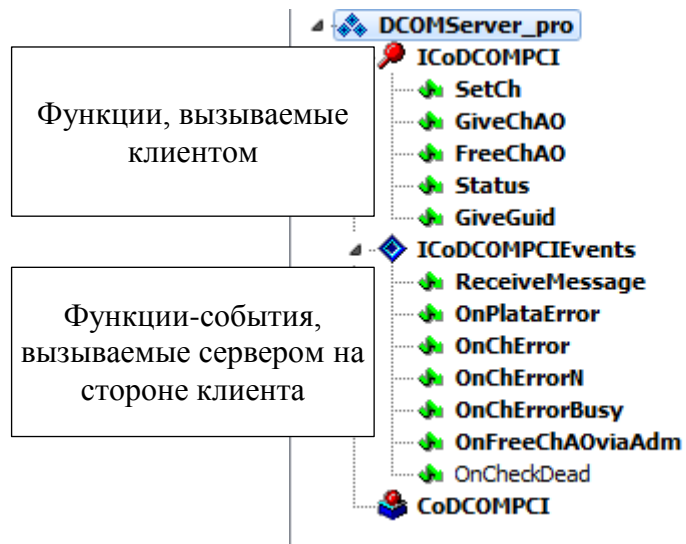


Рисунок 2 – Реализованные в библиотеке типов функции

Таблица 1 – Функции сервера, вызываемые клиентом

Название функций	Описание																					
SetCh	<p>Выдача на зарезервированные каналы платы напряжений. Перед вызовом функции, клиент должен иметь зарезервированными для записи каналы ЦАП сервера.</p> <p>Параметры¹:</p> <table border="1" data-bbox="467 457 984 751"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Modifier</th> </tr> </thead> <tbody> <tr> <td>_saiChN</td> <td>SAFEARRAY(int)</td> <td>[in]</td> </tr> <tr> <td>_safChVals</td> <td>SAFEARRAY(float)</td> <td>[in]</td> </tr> <tr> <td>iPlataN0</td> <td>int</td> <td>[in]</td> </tr> <tr> <td>_piMessage</td> <td>int*</td> <td>[out]</td> </tr> <tr> <td>_piPeerID</td> <td>int*</td> <td>[in, out]</td> </tr> <tr> <td>_guidCL</td> <td>struct GUID</td> <td>[in]</td> </tr> </tbody> </table> <p>_saiChN (SAFEARRAY(int)) – номера каналов, на которые необходимо выдать напряжения;</p> <p>_safChVals (SAFEARRAY(float)) – значения напряжений, которые необходимо выдать в вольтах (в диапазоне ± 10 В);</p> <p>iPlataN0 – номер платы на компьютере, на которую нужно выдать напряжение (нумерация плат начинается с «0»);</p> <p>_piMessage (int*) – указатель на область памяти, в которую будет записан текст сообщения от сервера клиенту (для преобразования номера сообщения в текст необходимо использовать файлы ALL\defines.h и ALL\defines.cpp);</p> <p>_piPeerID (int*) – в функции не используется, записан для сохранения общности структуры всех функций;</p> <p>_guidCL (guid) – идентификатор клиента, используется для проверки занятости запрашиваемых для записи напряжений каналов клиентом. Если хотя бы один из каналов занят другим клиентом – сервер вернет ошибку.</p>	Name	Type	Modifier	_saiChN	SAFEARRAY(int)	[in]	_safChVals	SAFEARRAY(float)	[in]	iPlataN0	int	[in]	_piMessage	int*	[out]	_piPeerID	int*	[in, out]	_guidCL	struct GUID	[in]
Name	Type	Modifier																				
_saiChN	SAFEARRAY(int)	[in]																				
_safChVals	SAFEARRAY(float)	[in]																				
iPlataN0	int	[in]																				
_piMessage	int*	[out]																				
_piPeerID	int*	[in, out]																				
_guidCL	struct GUID	[in]																				

¹ Параметры с модификатором [in] являются входными для функции, [out] – выходными, [in,out] – входными и выходными (значение, передаваемое по указателю функции, может быть изменено функцией)

Название функций	Описание																		
GiveChAO	<p>Запрос на занятие каналов платы клиентом. Если клиент ранее не был зарегистрирован на сервере, ему будет присвоен ID (<code>_piPeerID</code>). Если запрошен занятый канал, сервер вернет ошибку клиенту.</p> <p>Параметры:</p> <table border="1" data-bbox="495 472 1026 760"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Modifier</th> </tr> </thead> <tbody> <tr> <td><code>_saiChN</code></td> <td><code>SAFEARRAY(int)</code></td> <td><code>[in]</code></td> </tr> <tr> <td><code>iPlataNO</code></td> <td><code>int</code></td> <td><code>[in]</code></td> </tr> <tr> <td><code>_piMessage</code></td> <td><code>int*</code></td> <td><code>[out]</code></td> </tr> <tr> <td><code>_piPeerID</code></td> <td><code>int*</code></td> <td><code>[in, out]</code></td> </tr> <tr> <td><code>_guidCL</code></td> <td><code>struct GUID</code></td> <td><code>[in]</code></td> </tr> </tbody> </table> <p><code>_saiChN</code> (<code>SAFEARRAY(int)</code>) – номера каналов, которые необходимо зарезервировать; остальные параметры аналогичны по назначению параметров функции <code>SetCh</code>.</p>	Name	Type	Modifier	<code>_saiChN</code>	<code>SAFEARRAY(int)</code>	<code>[in]</code>	<code>iPlataNO</code>	<code>int</code>	<code>[in]</code>	<code>_piMessage</code>	<code>int*</code>	<code>[out]</code>	<code>_piPeerID</code>	<code>int*</code>	<code>[in, out]</code>	<code>_guidCL</code>	<code>struct GUID</code>	<code>[in]</code>
Name	Type	Modifier																	
<code>_saiChN</code>	<code>SAFEARRAY(int)</code>	<code>[in]</code>																	
<code>iPlataNO</code>	<code>int</code>	<code>[in]</code>																	
<code>_piMessage</code>	<code>int*</code>	<code>[out]</code>																	
<code>_piPeerID</code>	<code>int*</code>	<code>[in, out]</code>																	
<code>_guidCL</code>	<code>struct GUID</code>	<code>[in]</code>																	
FreeChAO	<p>Запрос на освобождение зарезервированных каналов. Функция, обратная функции <code>GiveChAO</code>.</p>																		
Status	<p>Функция передает клиенту количество каналов на платах, двумерный массив занятости каналов каждой платы</p> <table border="1" data-bbox="495 1144 1128 1432"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Modifier</th> </tr> </thead> <tbody> <tr> <td><code>_saiChN</code></td> <td><code>SAFEARRAY(__int64) *</code></td> <td><code>[out]</code></td> </tr> <tr> <td><code>_saiPlats</code></td> <td><code>SAFEARRAY(int) *</code></td> <td><code>[out]</code></td> </tr> <tr> <td><code>_piMessage</code></td> <td><code>int*</code></td> <td><code>[out]</code></td> </tr> <tr> <td><code>_piPeerID</code></td> <td><code>int*</code></td> <td><code>[in, out]</code></td> </tr> <tr> <td><code>_guidCL</code></td> <td><code>struct GUID</code></td> <td><code>[in, out]</code></td> </tr> </tbody> </table> <p><code>_saiPlats</code> (<code>SAFEARRAY(int) *</code>) – выходной массив, содержащий количество каналов плат; <code>_saiChN</code> – выходной массив, каждый элемент которого является словом, каждый бит которого соответствует каналу ЦАП, при этом если бит равен 1, то канал занят; если бит равен 0 – бит свободен.</p>	Name	Type	Modifier	<code>_saiChN</code>	<code>SAFEARRAY(__int64) *</code>	<code>[out]</code>	<code>_saiPlats</code>	<code>SAFEARRAY(int) *</code>	<code>[out]</code>	<code>_piMessage</code>	<code>int*</code>	<code>[out]</code>	<code>_piPeerID</code>	<code>int*</code>	<code>[in, out]</code>	<code>_guidCL</code>	<code>struct GUID</code>	<code>[in, out]</code>
Name	Type	Modifier																	
<code>_saiChN</code>	<code>SAFEARRAY(__int64) *</code>	<code>[out]</code>																	
<code>_saiPlats</code>	<code>SAFEARRAY(int) *</code>	<code>[out]</code>																	
<code>_piMessage</code>	<code>int*</code>	<code>[out]</code>																	
<code>_piPeerID</code>	<code>int*</code>	<code>[in, out]</code>																	
<code>_guidCL</code>	<code>struct GUID</code>	<code>[in, out]</code>																	
GiveGuid	<p>Функция генерирует <code>guid</code> (уникальный идентификатор клиента) на стороне сервера и передает клиенту (клиент может генерировать <code>guid</code> самостоятельно, без вызова серверной функции).</p>																		

Таблица 2 – События на стороне клиента, вызываемые сервером

Название функций	Описание												
ReceiveMessage	<p>Функция служит для передачи внутренних событий сервера клиентам.</p> <p>Parameters</p> <table border="1" data-bbox="553 436 1373 604"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Modifier</th> </tr> </thead> <tbody> <tr> <td>_piMessage</td> <td>int*</td> <td>[in, out]</td> </tr> <tr> <td>_piIntData</td> <td>int*</td> <td>[in, out]</td> </tr> <tr> <td>_fFloatData</td> <td>float*</td> <td>[in, out]</td> </tr> </tbody> </table> <p>Событие передает номер сообщения, а также дополнительные значения типа <i>int</i> и <i>float</i>.</p>	Name	Type	Modifier	_piMessage	int*	[in, out]	_piIntData	int*	[in, out]	_fFloatData	float*	[in, out]
Name	Type	Modifier											
_piMessage	int*	[in, out]											
_piIntData	int*	[in, out]											
_fFloatData	float*	[in, out]											
OnPlataError	<p>Внутренняя ошибка сервера при работе с платами: генерируется в случае запроса каналов неустановленной платы в компьютере (либо в случае недоступности платы со стороны штатного драйвера).</p> <p>Parameters</p> <table border="1" data-bbox="553 949 1373 1117"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Modifier</th> </tr> </thead> <tbody> <tr> <td>_iOldUncorPlataNO</td> <td>int</td> <td>[in]</td> </tr> <tr> <td>_saiAvailableNOs</td> <td>SAFEARRAY(int)</td> <td>[in]</td> </tr> <tr> <td>_piNewCorPlataNO</td> <td>int*</td> <td>[in, out]</td> </tr> </tbody> </table> <p>Передает клиенту номер запрошенной платы, доступные для занятия номера, и указатель, в который клиент должен записать номер новой запрашиваемой платы.</p>	Name	Type	Modifier	_iOldUncorPlataNO	int	[in]	_saiAvailableNOs	SAFEARRAY(int)	[in]	_piNewCorPlataNO	int*	[in, out]
Name	Type	Modifier											
_iOldUncorPlataNO	int	[in]											
_saiAvailableNOs	SAFEARRAY(int)	[in]											
_piNewCorPlataNO	int*	[in, out]											
OnChError	<p>Событие генерируется в случае, если запрошен канал номером больше, чем есть на плате.</p> <p>Parameters</p> <table border="1" data-bbox="553 1411 1373 1579"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Modifier</th> </tr> </thead> <tbody> <tr> <td>_iOldUncorChN</td> <td>int</td> <td>[in]</td> </tr> <tr> <td>_saiAvailableChNs</td> <td>SAFEARRAY(int)</td> <td>[in]</td> </tr> <tr> <td>_piNewCorChN</td> <td>int*</td> <td>[in, out]</td> </tr> </tbody> </table> <p>Передает клиенту номер запрошенного канала, номера доступных каналов, указатель на запись нового запрашиваемого канала.</p>	Name	Type	Modifier	_iOldUncorChN	int	[in]	_saiAvailableChNs	SAFEARRAY(int)	[in]	_piNewCorChN	int*	[in, out]
Name	Type	Modifier											
_iOldUncorChN	int	[in]											
_saiAvailableChNs	SAFEARRAY(int)	[in]											
_piNewCorChN	int*	[in, out]											

OnChErrorBusy	<p>Событие генерируется, если запрашиваемый канал занят другим клиентом.</p> <p>Return Type: <input type="text" value="HRESULT"/></p> <p>Parameters</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Modifier</th> </tr> </thead> <tbody> <tr> <td>_OldUncorsaiChN</td> <td>SAFEARRAY(int)</td> <td>[in]</td> </tr> <tr> <td>_saiAvailableChNs</td> <td>SAFEARRAY(int)</td> <td>[in]</td> </tr> <tr> <td>_psaiNewCorsaiChN</td> <td>SAFEARRAY(int) *</td> <td>[in, out]</td> </tr> </tbody> </table> <p>Параметры аналогичны событию OnChError.</p>	Name	Type	Modifier	_OldUncorsaiChN	SAFEARRAY(int)	[in]	_saiAvailableChNs	SAFEARRAY(int)	[in]	_psaiNewCorsaiChN	SAFEARRAY(int) *	[in, out]
Name	Type	Modifier											
_OldUncorsaiChN	SAFEARRAY(int)	[in]											
_saiAvailableChNs	SAFEARRAY(int)	[in]											
_psaiNewCorsaiChN	SAFEARRAY(int) *	[in, out]											
OnFreeChAOviaAdm	<p>Зарезервированное событие для клиента (в текущей версии сервера не реализован вызов сервером) в случае, если канал освобождается пользователем из программы сервера (освобождение канала администратором).</p>												
OnCheckDead	<p>Зарезервированное событие для будущих релизов</p>												

При вызове каждой функции и после ее завершения для безошибочной работы с множеством клиентов программа-сервер блокирует и разблокирует критическую секцию (*EnterCriticalSection / LeaveCriticalSection*), создаваемую программой при запуске. Выполняется с целью синхронизации работы всех клиентов.

Множественный доступ к платам (если в компьютере установлено более одной платы *1724U*) осуществляется вызовом соответствующих объектов вектора *vectInstantAoCtrls* на стороне сервера.

Таким образом процедура работы с сервером со стороны клиента выглядит следующим образом:

1. Получение клиентом индивидуального номера (*GUID*) самостоятельно или при помощи сервера (*GiveGuid*);
2. Запрос состояния сервера вызовом команды *Status*, результатом выполнения которой является передача клиенту информации о количестве доступных плат и каналов;
3. Запрос на предоставление одного или нескольких каналов платы;
4. Непосредственной реализации управления путем выдачи управляющих напряжений на занятые каналы платы командами *SetCh*;
5. Освобождения канала клиентом на стороне сервера (*FreeChAO*).

Для удобства пользователей клиента на стороне клиента должны быть реализованы обработчики событий, вызываемые сервером (таблица 2).