# Описание компонентов модулей фирмы LCard

Программный компонент LCardE140 модуля E14-140 фирмы LCard предназначен для использования в среде разработчика Delphi, что позволяет резко сократить время на разработку приложений, применяющих данный модуль. Компонент LCardDevicesList делает обзор подключенных к компьютеру LCard модулей и позволяет выбрать один из них.

1. **Инсталляция компонентов.**

Создайте каталог LCard и распакуйте туда файлы из LCard.zip. Обновите, при необходимости, библиотеку Lcomp и драйверы X502api.dll, L502api.dll, E502api.dll (lpcie\_setup.exe) с официального сайта. Для того, чтобы в палитре компонентов Delphi появились указанные компоненты необходимо в меню оболочки <Component> -> <Install Packages> нажать кнопку Add и выбрать пакет LCard.bpl. Компоненты появятся в разделе “Модули фирмы LCard” палитры компонентов.

1. **Компонент LCardE140.**

Компонент содержит базовые свойства, методы и события, а также свойства АЦП (ADC) и ЦАП (DAC), их настройка определяет функционал и режимы работы модуля.

***Опубликованные базовые свойства компонента:***

* AboutComponent: string – содержит номер версии п.о. компонента;
* ADC – свойства АЦП, подробно описаны ниже;
* DAC – свойства ЦАП, подробно описаны ниже.

***Неопубликованные базовые свойства компонента:***

* Connect: boolean – флажок наличия соединения модулем;
* SerNum: string – серийный номер модуля;
* BrdName: string – наименование модуля;
* Rev: string – ревизия модуля;
* DspType: string – тип DSP;
* IsDacPresent: boolean – флаг наличия ЦАП;
* Quartz: cardinal – частота кварца;
* Slot: integer – номер логического слота модуля;
* DIGInput: array[0..15] of boolean – массив состояний дискретных входов;
* DIGOutput: array[0..15] of boolean – массив состояний дискретных выходов;
* DIGOutputEnable: boolean – блокировка дискретных выходов;
* StatusLCard: word - текущее состояние (режим работы) модуля, биты StatusLCard:

D0 – “1”- режим циклического чтения из АЦП;

D1 – “1” – режим работы ЦАП в циклическом или потоковом режиме;

D2 – “1” - состояние ожидания срабатывания условия начала чтения из АЦП.

***Опубликованные свойства ADC:***

* ADCControlCount: cardinal – определяет количество опрашиваемых каналов АЦП;
* ControlTable: string – содержит список и очередность опроса логических каналов АЦП. Для вызова диалогового окошка формирования таблицы логических каналов необходимо нажать кнопочку <…> в конце строки или дважды кликнуть в строке списка. В диалоговом окошке двойной клик мышкой или нажатие кнопки «Enter» в требуемой ячейке вызывает появление выпадающего списка допустимых параметров. Для фиксации выбранного значения необходимо нажать кнопку «Enter», нажатие «Esc» приведет к восстановлению исходного значения. Для сохранения таблицы логических каналов в свойстве ControlTable необходимо нажать кнопку «Сохранить»;
* Cycle: boolean – устанавливает циклический режим измерений;
* Frequency: double – определяет частоту измерений каналов АЦП в кгц;
* FormatData – назначает тип результатов замеров (V-вольты, mV-милливольты, Code-коды АЦП);
* SynchroParam – настраивает источник, тип и режим синхронизации измерений;

- NoSync – отсутствие синхронизации;

- TTLStartSync – TTL-синхронизация по заднему фронту на входе Int;

- TTLKadrSync - TTL-синхронизация по заднему фронту на входе Int для фиксации одного кадра замеров;

- AnalogSync – аналоговая синхронизация по значению входного напряжения на канале, заданном в поле SyncLogicalChannel;

- LevelSens – по уровню;

- HigherLevelDownToUp – по уровню выше;

- LowerLevelTopToDown – по уровню ниже;

- TransitionSens – по переходу;

- HigherLevelDownToUp – по переходу снизу-вверх;

- LowerLevelTopToDown – по переходу сверху-вниз;

- ExtClock – настраивает модуль на внешний источник тактовых импульсов;

- ClockTrans – подключает внутренний генератор тактовых импульсов к выходу Syn;

- SyncPorog: smallint – пороговое напряжение на канале SyncLogicalChannel (мв) при аналоговой синхронизации.

***Неопубликованные свойства ADC:***

* ADCMeasuringCount: cardinal – задание количества кадров измерений. Запись ненулевого значения автоматически инициирует процесс накопления замеров;
* ADCMeasuringTime: cardinal – задание временного интервала (мс), в течение которого будет осуществляться накопление замеров. Запись ненулевого значения автоматически инициирует процесс накопления замеров;
* ADCSensors: array[0..31] of array of double - двумерный динамический массив замеров по каждому физическому каналу в режиме измерений относительно общей земли. Тип данных в массиве определяется свойством FormatData;
* ADCDiffSensors: array[0..15] of array of double - двумерный динамический массив замеров по каждому физическому каналу в режиме дифференциального измерения. Тип данных в массиве определяется свойством FormatData;
* MeasurementPeriod: double – период опроса каналов (мс) (рассчитывается модулем, исходя из значения поля Frequency), удобно использовать при построении графиков замеров. Свойство доступно только для чтения;
* StopScanADCData: boolean – флажок, блокирующий запись новых измерений в массивы ADCSensors и ADCDiffSensors до тех пор пока не будут обработаны предыдущие. Устанавливается автоматически перед активизацией события OnReadData, после завершения обработки замеров необходимо выполнить StopScanADCData:=false;
* Calibration0: boolean – флажок включает режим калибровки нуля модуля. Режим запускается после инициирования процесса накопления замеров (ADCMeasuringCount или ADCMeasuringTime);
* MeasureFileName: string – имя файла (полный путь) для сохранения/восстановления измерений в формате кодов АЦП, сам файл на момент записи может не существовать, в этом случае он будет создан автоматически;
* FlagStoreMeasureToFile: boolean – флажок включает режим записи измерений в файл. Процесс записи начинается после задания ненулевых значений параметрам ADCMeasuringCount или ADCMeasuringTime. Флаг автоматически сбрасывается по завершении записи в файл;
* FlagReStoreMeasureFromFile – флажок включает режим восстановления измерений из файла заданного в поле MeasureFileName. Необходимо иметь в виду, что значения свойств ADCControlCount, ControlTable в момент восстановления должны быть такими же как при записи. Данный режим удобно использовать для отладки программного обеспечения вне реальной системы, например в лаборатории, в спокойной обстановке. Флаг автоматически сбрасывается по завершении считывания измерений из файла.

***Методы ADC:***

* Procedure StopADCStream – вызов процедуры прерывает накопление данных (замеров);
* Function ADCASyncSensor(PhysCh, VoltageCoef: word; DiffMode: boolean): double – функция возвращает напряжение на физическом канале PhysCh, где VoltageCoef – коэффициент усиления входного напряжения (0..3), а DiffMode режим измерения (false - относительно общей земли, true – дифференциальное измерение). Тип данных определяется свойством FormatData;
* Procedure SetSyncLogicalChannel(PhysCh, VoltageCoef: word; DiffMode: boolean) – процедура формирует логический номер канала АЦП, используемого для синхронизации (SynchroParam). Параметры процедуры имеют тот же смысл, что и в функции ADCASyncSensor.

***Опубликованные свойства DAC:***

* FormatData – назначает тип выходных данных (V-вольты, mV-милливольты, Code-коды ЦАП);
* FrequencyDiv: word - делитель частоты дискретизации каналов ЦАП (допустимые значения делителя: 1..8);
* SizeBuffer: cardinal – размер буфера ЦАП;
* StopConstDAC0, StopConstDAC1: integer – напряжение (мв) на выходах DAC0(DAC1) после прерывания вывода на ЦАП в циклическом и потоковом режимах;
* StepCount: cardinal - количество интервалов подкачки в потоковом режиме, при этом необходимо учитывать что, буфер ЦАП (SizeBuffer) должен содержать целое количество интервалов подкачки для каждого канала (подробнее см. описание события OnDACStepSwapping).

***Неопубликованные свойства DAC:***

* RealFrequency: double - реальная частота дискретизации каналов ЦАП (рассчитывается модулем, исходя из значения поля FrequencyDiv). Свойство доступно только для чтения;
* DACData: array[0..1] of array of double – двумерный динамический массив выходных данных ЦАП для циклического и потокового режимов. Размер массива автоматически устанавливается компонентой в соответствии со значением поля SizeBuffer. Массив должен формироваться приложением значениями выходных данных ЦАП с учетом значения поля FormatData для достижения требуемой формы выходного сигнала;
* DACAsyncData: array[0..1] of double – массив выходных данных ЦАП для асинхронного режима выдачи. Данные должны записываться в массив с учетом значения поля FormatData.

***Методы DAC:***

* Procedure DACASync2Channels(DAC0,DAC1: double) – асинхронный вывод сразу на два канала ЦАП (только для ревизии B);
* Procedure StartDACStream - вывод данных в циклическом и потоковом режимах на оба канала ЦАП (только для ревизии B);
* Procedure StopDACStream – вызов процедуры прерывает вывод данных в циклическом и потоковом режимах.

***Базовые методы компонента:***

* Procedure WriteDigOutput - установка дискретных выходов в асинхронном режиме. Состояния выходов предварительно формируются в массиве DIGOutput;
* Function TextError(CodError: TErrorLCard): string - преобразует код ошибки в текстовую строку;
* Procedure CloseDevice - останов и закрытие измерительного модуля, при этом вырабатывается событие OnDisConnect.

***События компонента:***

* OnConnect(Sender: TObject) – событие активируется в момент соединения компьютера с модулем;
* OnDisConnect(Sender: TObject) - событие активируется в момент рассоединения компьютера с модулем. В этом случае, если предварительно не выполнялись останов и закрытие модуля, компонент будет в автоматическом режиме делать попытки восстановления соединения с модулем, имеющим серийный номер на момент последнего соединения;
* OnReadData(Sender: TObject) – событие завершения сбора заданного количества измерений. Количество замеров для каждого логического канала определяется как Length(ADCSensors[0..31]) или Length(ADCFreeSensors[0..15]). После завершения обработки данных необходимо выполнить присваивание ADC.StopScanADCData:=false;
* OnError(Sender: TObject; CodError: longint) – событие фиксации ошибки в работе с модулем, при этом переменная CodError будет содержать код ошибки. Преобразование кода ошибки в текст можно выполнить вызовом функции TextError(CodError): string;
* OnDigInputChange(Sender: TObject) - событие изменения состояния входов. Входы сканируются компонентом с периодом ~ 1 мс;
* OnDACStepSwapping(Sender: TObject; NumStepSwapping: ULong) - событие подкачки массива ЦАП, целесообразно использовать в потоковом режиме работы ЦАП, когда требуется динамическое изменение напряжения (формы сигнала) на выходах относительно предыдущего цикла. В событии передается переменная NumStepSwapping - номер части массива выходных данных канала, которую приложение может безопасно скорректировать. Количество частей, на которые может быть разбит массив, определяется в поле DAC. StepCount;
* OnAlgoritm(Sender: TObject) - событие реализации алгоритма управления. Вспомогательное событие, активируется с периодом ~ 1 мс, позволяет на основе данных с АЦП и дискретных входов создать систему управления объектом.

***Типы ошибок компонента:***

В процессе инициализации и работы компонента возможно появление тех или иных ошибок. Удобнее всего реагировать на возникновение ошибок в обработчике события OnError компонента. Событие возвращает код ошибки CodError: longint. TErrorLCard(CodError) включает в себя следующие типы ошибок:

* erNotLCompDll - не найдена библиотека "lcomp.dll";
* erInitParam - ошибка инициализации модуля;
* erStartLDevice - ошибка запуска модуля;
* erNotSupported - режим не поддерживается модулем;
* erDACAsync - ошибка ЦАП в асинхронном режиме (возникает при попытке установить значение на выходах ЦАП при работе в циклическом или потоковом режиме);
* erADCAsync - ошибка АЦП в асинхронном режиме (возникает при попытке считать значение на входах АЦП при работе в циклическом режиме или в состоянии ожидания срабатывания условия начала чтения АЦП).

1. **Компонент TLCardDevicesList.**

Компонент LCardDevicesList является потомком комбинированного списка TComboBox. С периодом в 1 секунду компонент делает обзор подсоединенных к компьютеру, но не подключенных к приложениям, LCard модулей и позволяет выбрать один из них.

***Опубликованные свойства компонента:***

* AboutComponent: string – содержит номер версии п.о. компонента;
* DevicesFilter: TDevicesFilter - фильтр выборки LCard модулей, позволяет построить список (Items) только из необходимых модулей, использование в качестве фильтра значения dfAnyLCardDevice приведет к выборке всех доступных модулей. Строка списка содержит тип модуля, его серийный номер и логический номер слота.

***Неопубликованные свойства компонента:***

* SerNum: string – серийный номер выбранного из списка модуля;
* Slot: integer – логический номер слота выбранного из списка модуля;
* BrdName: string - название выбранного из списка модуля.