

ВИКОРИСТАННЯ ДОДАТКОВИХ БІБЛІОТЕК В СЕРЕДОВИЩІ ПРОГРАМУВАННЯ CODESYS 2.3 З МЕТОЮ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПРОГРАМОВАНИХ КОНТРОЛЕРІВ

*Лісовець С.М. – к.т.н., ser.lis.290171@gmail.com
Київський національний університет технологій та дизайну*

Проведене дослідження програмованих контролерів компанії ОВЕН і середовища їх програмування CoDeSys 2.3 показало, що використання додаткових бібліотек дозволяє суттєво полегшити процес програмування таких контролерів, а вихідний програмний код робить більш прозорим і зрозумілим.

A study of OVEN programmable controllers and their CoDeSys 2.3 programming environment showed that the use of additional libraries can greatly facilitate the programming process of such controllers, and the source code makes it more transparent and understandable.

Вступ. Майже всі сучасні технологічні процеси в різних галузях виробництва потребують застосування засобів обчислювальної техніки. Такі засоби дозволяють виконувати практично всі функції, які стосуються показу, сигналізації, реєстрації і регулювання технологічних параметрів. В якості засобів обчислювальної техніки можуть використовуватися промислові комп'ютери в різному виконанні (настінні і настільні, панельні, вбудовані, переносні тощо), програмовані контролери, окремі модулі і плати збору даних, прилади на жорсткій логіці (регулятори, лічильники, таймери тощо).

Найбільш часто використовуються програмовані контролери. З одного боку, вони мають функціональні можливості не гірше, ніж у промислових комп'ютерів. З іншого боку, вони мають надійність не гірше, ніж у приладів на жорсткій логіці. Зазвичай програмовані контролери мають або вбудоване середовище виконання (прошивку), або вбудовану операційну систему. Середовище виконання забезпечує більшу надійність виконання програмного коду, а операційна система – більшу функціональність. Тому для роботи безпосередньо з обладнанням використовуються контролери з середовищем виконання, а для подальшої ефективної обробки даних – контролери з операційною системою.

Одним з основних виробників програмованих контролерів, а також іншого обладнання для автоматизації в Україні є компанія ОВЕН

(м. Харків). Вона випускає програмовані контролери ПЛК63, ПЛК73, ПЛК100, ПЛК150, ПЛК154 і деякі інші, які дозволяють працювати безпосередньо з обладнанням і мають вбудоване середовище виконання, а програмуються за допомогою середовища програмування CoDeSys 2.3. Програмування здійснюється згідно з вимогами МЕК 61131-3 [1-4].

Постановка проблеми. Для програмування програмованих контролерів зазвичай застосовуються одна або кілька мов (IL, ST, SFC, LD, FBD, CFC) і стандартні бібліотеки `standard.lib` і `util.lib`. При цьому часто недостатньо уваги приділяється додатковим бібліотекам, які також входять до складу поставки контролерів: такі бібліотеки суттєво розширюють їх функціональні можливості і значно спрощують процес програмування. Таким чином, була поставлена задача дослідити додаткові можливості програмування середовища програмування CoDeSys 2.3.

Результати досліджень. Для обміну даними із “зовнішнім світом” програмовані контролери використовують різні інтерфейси: RS-232, RS-485, USB, Ethernet, CAN і деякі інші. З таких інтерфейсів можна виділити, зокрема, інтерфейс RS-485, основне призначення якого полягає у створенні комп’ютерних мереж у виробничих умовах. Багато засобів автоматизації взагалі мають тільки один інтерфейс RS-485, і обмін даними з ними можна здійснювати тільки за допомогою цього інтерфейсу. Тому для роботи з інтерфейсом RS-485 можна використати бібліотеку `SysLibCom.lib`. Відкриття і закриття порту здійснюється за допомогою функцій `SysComOpen` і `SysComClose`, налаштування порту – за допомогою функції `SysComSetSettings`, а передача і прийом даних – за допомогою функцій `SysComWrite` і `SysComRead`. Особливістю цієї бібліотеки є наявність функції `SysComSetSettingsEx`, як дозволяє контролювати потік даних і визначати кількість біт, які описують в потоці даних один символ.

Одним з напрямів розвитку засобів автоматизації є перехід на більш швидкісні інтерфейси, зокрема, на інтерфейс Ethernet. Такий інтерфейс передбачає використання протоколу мережевого рівня IP та протоколів транспортного рівня UDP і TCP. Тому для роботи з інтерфейсом Ethernet можна використати бібліотеку `SysLibSockets.lib`, яка базується на програмному інтерфейсі “сокет”. Відкриття і закриття “сокета” здійснюється за допомогою функцій `SysSockCreate` і `SysSockClose`, налаштування “сокету” – за допомогою функції `SysSockSetOption`, надання “сокету” IP-адреси – за допомогою функції `SysSockSetIPAddress`, встановлення “сокетом” з’єднання – за допомогою функції `SysSockConnect`, прослуховування “сокетом” з’єднання – за допомогою функції `SysSockListen`, прийняття “сокетом” з’єднання – за допомогою функції `SysSockAccept`, вибір “сокета” – за допомогою функції

SysSockSelect, а передача і прийом даних – за допомогою функцій SysSockSendTo і SysSockRecvFrom (по протоколу UDP) та SysSockSend і SysSockRecv (по протоколу TCP).

Більшість існуючих на теперішній час програмованих контролерів можуть виконувати не одну, а кілька задач одночасно. Це дає змогу за допомогою одного програмованого контролера підтримувати роботу кількох контурів керування (наприклад, одночасно керувати температурою і рівнем рідини в ємності). Такі контури керування можуть працювати абсолютно незалежно, і закон підтримання температури рідини в ємності може ніяк не залежати від рівня рідини в цій ємності. Але часто технологічні операції залежать одна від одної: наприклад, відбирати рідину з ємності можна тільки тоді, коли її температура досягла заданого значення. Такі технологічні операції повинні бути синхронізованими в часі, що досягти за допомогою стандартних прийомів програмування досить складно. Тому для роботи з кількома задачами, які потрібно синхронізувати одна з одною, можна використати бібліотеку SysLibEvent.lib. Синхронізація виконується шляхом створення, активації і видалення відповідних подій. Така синхронізація може бути виконана, в загальному випадку, між двома будь-якими задачами: одна із задач не буде виконуватися і при цьому буде чекати події, яка може бути (а може і не бути) активована іншою задачею. Створення події здійснюється за допомогою функції SysEventCreate, активація події – за допомогою функції SysEventSet, задання тайм-ауту – за допомогою функції SysEventWait, а видалення події – за допомогою функції SysEventDelete.

Для роботи із задачами також можна використати бібліотеку SysLibTasks.lib. Така бібліотека дозволяє створювати, видаляти, запускати і зупиняти задачі, а також змінювати їх пріоритет. Таким чином, з'являється можливість повністю контролювати порядок виконання технологічних операцій. Зокрема, для створення задачі з пріоритетом від 0 до 255 використовується функція SysTaskCreate, для видалення задачі – функція SysTaskDestroy, для призупинки і перезапуску задачі – функції SysTaskSuspend і SysTaskResume, для призупинки задачі на заданий час – функція SysTaskSleep.

Також важливим є розміщення безпосередньо в накопичувачі програмованого контролера різних файлів. Наприклад, це можуть бути файли, які містять вихідний програмний код, номінальну статичну характеристику вимірювального перетворювача, калібрувальні коефіцієнти і так далі. Часто наявність таких файлів вимагає замовник автоматизованої системи. Для роботи з файлами можна використати бібліотеку SysLibFile.lib. В цій бібліотеці для відкриття (створення) і закриття файлів

використовуються функції SysFileOpen і SysFileClose, для видалення файлів – функція SysFileDelete, для обміну даними з файлами – функції SysFileWrite і SysFileRead.

Іноді виникає потреба у безпосередньому керуванні роботою програмованого контролера. Для цього використовується бібліотека SysLibPLCCtrl.lib. Шляхом виклику функції SysStartPLCProgram з цієї бібліотеки програмований контролер можна запустити, шляхом виклику функції SysStopPLCProgram – зупинити, шляхом виклику функції SysResetPLCProgram – перезавантажити, а шляхом виклику функції SysShutdownPLC – вимкнути.

Крім того, не останню роль грає і обробка переривань. Застосування переривань значно зменшує навантаження на обчислювальну частину програмованого контролера, так як для обробки події, яка викликала переривання, необхідно тільки написати програмний код як “реакцію” на цю подію. Для цього використовується бібліотека SysLibInt.lib. Функція SysInstallIntHandler цієї бібліотеки дозволяє зв’язати оброблювач переривань заданої події з певною функцією, а функція SysRemoveIntHandler дозволяє розірвати цей зв’язок.

Необхідно зауважити, що крім розглянутих бібліотек середовища програмування CoDeSys 2.3, які застосовуються найчастіше, існують також бібліотеки для роботи з каталогами файлової системи, для доступу до змінних на основі прямих адрес, для керування пам’яттю, для роботи з годинниками реального часу і деякі інші. Використання таких бібліотек значно полегшує процес програмування програмованих контролерів, а вихідний програмний код робить більш прозорим і зрозумілим.

Висновки. Використання при створенні програмного коду в середовищі програмування CoDeSys 2.3 додаткових бібліотек дозволяє суттєво розширити функціональні можливості програмованих контролерів.

Список використаних джерел

1. Деменков Н.П. Языки программирования промышленных контроллеров: Учебное пособие / Под. ред. К.А. Пупкова. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. – 172 с.: ил. – ISBN 5–7038–2608–Х.
2. Ельперін І.В. Промислові контролери: Навчальний посібник / І.В. Ельперін – К.: НУХТ, 2003. – 320 с. – ISBN 966–612–024–0.
3. Ключев А.С. Автоматизация настройки систем управления / А.С. Ключев, В.Я. Ротач, В.Ф. Кузицин. – М.: Альянс, 2015. – 272 с. – ISBN 978–5–91872–091–2.
4. Минаев И.Г. Свободно программируемые устройства в автоматизированных системах управления / И.Г. Минаев, В.В. Самойленко, Д.Г. Ушкур, И.В. Федоренко. – Ставрополь: АГРУС, 2016. – 168 с. – ISBN 978–5–9596–1222–1.