

ПРОГРАММА УПРАВЛЕНИЯ ШИРОТНО-ИМПУЛЬСНЫМ ПРЕОБРАЗОВАТЕЛЕМ

Петухов В.С., 2 курс,

Новиков С.О., ст.преподаватель,

УО «Белорусский национальный технический университет»

В настоящей работе рассматривается создание программного обеспечения для системы управления, а также освещаются некоторые преимущества среды разработки и платформы под которую ведётся разработка.

Программируемые логические контроллеры (ПЛК), устройства, одной из главных особенностей которых является возможность использовать их практически везде, где требуется осуществлять управление каким-либо процессом лишь загрузив в них соответствующее программное обеспечение. Одной из лучших сред для программирования ПЛК является среда исполнения *Controllers Development System (CoDeSys)*, средства разработки и отладки которой базируются на уже существующих в других широкоизвестных средах программирования. Целью данной работы является разработка программного обеспечения для контроллера с использованием исключительно возможностей *CoDeSys* без использования какого-либо стороннего оборудования, что должно показать преимущества как среды разработки, так и самого контроллера.

Разработанное программное обеспечение предназначено для управления широтно-импульсным преобразователем (ШИП), устройством, предназначенным для периодического дозирования потока энергии передаваемого от источника к потребителю. Преимущество такого многоцелевого контроллера, как ПЛК состоит в том, что способы импульсного регулирования бывают разные, но при этом они похожи между собой (так как производятся манипуляции с одними и теми же параметрами). В программе осуществляется регулирование выходных импульсов: их длительности (в программе – переменное значение *tt*) и периода следования (в программе – постоянное значение, *Period.PT*, равно 5 секундам). Проект состоит из двух *Program Organization Unit (POU)* типа «программа» написанных на языках *CFC, SFC, ST*.

Первый *POU (PLC_PRG)* является главным и написан на языке *CFC*, графическом редакторе, который позволяет наглядно представить работу некоторых компонентов (например таймеров) и взаимосвязей в программе. В нём расположены два таймера один из которых (*Period*) отмеряет период следования импульсов, а другой (*Tau*) – их длительность. Состояния стоящие у них на выходах переменные *Res1* и *Res2* определяют наличие или отсутствие сигнала. Но как было сказано ранее, продолжительность импульсов должна определённым образом меняться. Если бы выходные сигналы можно было бы представить в виде графика, то он был бы похож на параболу, некоторые фрагменты которой вырезаны (причем фрагменты неравной длины). Этим преобразованием занимается вторая программа (*TI*) которая вызывается в этом же блоке.

POU TI написан с использованием двух языков программирования *SFC* и *ST*. *SFC* – язык позволяющий описать последовательность различных действий в программе. Причём сами действия описываются на другом языке (в данном случае – *ST*). Алгоритм разветвляется на четыре ветви, выполняемые параллельно. Первые три ветви содержат по одному блоку. Блок *StartI* запускает таймер *Period*, блок *Graph* осуществляет предварительную подготовку выходного сигнала. Блок *GraphG* реагирует на включения и выключения таймера *Tau*, формируя окончательный вид сигнала. На какой промежуток времени должен включаться этот таймер и как долго он должен оставаться выключенным определяет последняя ветвь. Она состоит из одиннадцати блоков. В блоках *T200, T400, T600, T800, T1000* происходит установка значения *tt* (при возрастании графика длительность импульсов увеличивается с шагом 200, а при убывании – уменьшается (с тем же шагом)) и запуск таймера *Tau*. В блоках *ABORT1, ABORT2, ABORT3, ABORT4, ABORT5* таймер приводится в готовность к повторному запуску. В блоке *INV* определяется возрастает график или убывает.

Большим плюсом среды *CoDeSys* является то, что для проверки работоспособности программы не обязательно наличие какого-либо дополнительного оборудования в том числе и ПЛК. Проверку данной программы удобно провести с использованием режима трассировки, который в реальном времени построит график работы.

Таким образом, в результате проделанной работы было разработано программное обеспечение для управления широтно-импульсным преобразователем, причём для этого не потребовалось никакого дополнительного оборудования, кроме персонального компьютера с установленной на нём средой разработки, что объясняется унифицированностью ПЛК.