

Здравствуйте, Анатолий Абрамович!

Я написал программу проведения поверки преобразователя массового расхода (ПМР) по однонаправленной трубопоршневой поверочной установке (ТПУ) для контроллера *TERN*.

Программу спроектировал, используя автоматное программирование. Граф переходов автомата с учетом анализа на выбросы содержал шесть состояний. Его я перевел в программу на языке C++, которую отладил, подавая на импульсный вход частоту с калибратора и на дискретный вход, выдающий прерывание, сигнал напряжения с аппаратного имитатора детекторов ТПУ. Однако по независящим от меня причинам заниматься этим контроллером перестали.

Мне дали работу – реализация пользовательских требований по учету нефти на базе другого контроллера (было предложено использовать его как контроллер расхода для нефти). Я перенес свою программу на этот контроллер, внося необходимые корректировки из-за новой идеологии, среды разработки, файловой системы и изменившихся требований (удален анализ на выбросы).

Попутно нашел временно свободный контроллер *Omni* (нет ничего более постоянного, чем что-нибудь временное). Подключил его дискретный вход для детекторов ТПУ и импульсный вход для импульсов расхода параллельно соответствующим входам *другого контроллера*. Затем подал на них сигналы с имитатора детекторов и частоту с калибратора. Кроме того, к каждому контроллеру подключен компьютер. Итак, получил отличный полигон.

На нем имитировал поверку по ТПУ. Результаты, полученные на обоих контроллерах одновременно, меня удовлетворили. Потом неожиданно для меня поступило требование реализовать поверку по мастер-счетчику – контрольному расходомеру. Граф переходов для ее проведения оказался похож на граф переходов проведения поверки по ТПУ, и я получил его из этого графа корректировкой в *MS Visio*. Перевел его в автоматную программу на языке C++, причем не стал заморачиваться с выделением отдельного класса для мастер-счетчика, так как слаб в объектно-ориентированном программировании и считал, что готово должно быть еще вчера. Поэтому в необходимых местах ввел оператор типа

```
switch (type_prover)
{
    case Unidirect:
        ...
        break;
    case Master:
        ...
        break;
}
```

Для того чтобы протестировать случай `case Master`, в полигоне прогнал поверку по мастер-счетчику. Ее основным результатом является М-фактор (отношение массы, измеренной мастер-счетчиком, к массе по поверяемому ПМР). На *Omni* и *другом контроллере* М-факторы совпали, массы были близки. Пришла в голову мысль вставить в граф переходов еще одно состояние «Выполняется переход» к следующей точке расхода. Цель – дать время машинисту насоса изменить обороты электродвигателя и продолжить поверку, когда будет обеспечен требуемый расход.

Граф переходов получился немного кривой, так как новое состояние пририсовал сбоку, и ветви пересекались. Допустил ошибку, из-за которой программа зацикливалась между двумя состояниями. Отредактировал граф и затем программу. Итак, для проверки по мастер-счетчику получил граф переходов с шестью состояниями. Результат прогона теста в полигоне был положителен. Используя граф переходов и схему связей автомата, я быстро и четко ответил на вопрос начальника.

Спасибо, что сайт <http://is.ifmo.ru> и Ваша книга «Автоматное программирование» помогают мне самостоятельно идти к цели работы.

С уважением, Аркадий Лившиц. 04.12.12.