

УДК 62.05

РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ ОПТИМИЗАЦИИ ПРОЦЕССА ДИАГНОСТИРОВАНИЯ ПЕЧАТНЫХ ПЛАТ С ПРОГРАММИРУЕМЫМ МИКРОКОНТРОЛЛЕРОМ

Шадрин В.В. (федеральное государственное бюджетное образовательное учреждение высшего образования «Иркутский государственный университет»),

Научные руководители – Веснин А.М., Громик Н.А.

(федеральное государственное бюджетное образовательное учреждение высшего образования «Иркутский государственный университет», федеральное государственное бюджетное учреждение науки Ордена Трудового Красного Знамени Институт солнечно-земной физики Сибирского отделения Российской академии наук)

Введение.

При разработке программируемых печатных плат есть две составляющие: программа (прошивка) микроконтроллера и аппаратная обвязка. Программа должна учитывать аппаратные особенности, одним из простых примеров которой является дребезг тактовых кнопок. Поэтому тестирование прошивки отличается от тестирования, к которому мы привыкли в разработке программного обеспечения (юнит-тестирование, для примера). Прошивка должна работать правильно с учетом тех элементов, которые находятся на печатной плате. Если это не так, часто бывает возможным поправить поведение программно, без изменения компонентов, но иногда приходится заменять одни элементы на другие (менять конденсаторы на модели с большей ёмкостью [1], для примера). Поэтому важно иметь инструмент, который поможет протестировать не только код, но и взаимодействие этого кода с железом. Это возможно сделать, используя эталонные (модельные) отклики программно-аппаратного комплекса и реальные отклики, которые мы получаем на реальной печатной плате. Цель данной работы - разработка программного обеспечения для оптимизации процесса диагностирования печатных плат с программируемым микроконтроллером.

Основная часть.

Мы разрабатываем приложение для настольного компьютера, которое будет связываться с одним или двумя микроконтроллерами для отправки и/или считывания с них данных. Эталонным сценарием работы считается тот, при котором на один микроконтроллер отправляются команды для выполнения тестируемым устройством, а второй микроконтроллер считывает сигналы с данного устройства и сохраняет их для дальнейшего анализа. После завершения сеанса отправки-считывания, приложение сравнивает полученные сигналы с эталонными, - если таковые были предоставлены пользователем заранее, - и выдаёт наглядную таблицу с результатами тестирования, которую можно сохранить в отдельный файл или же сразу занести в поддерживаемую приложением базу данных.

Пошаговый принцип работы:

- Пользователь указывает конфигурационный файл с ключами и значениями, где “q” – запрос, “a” – ожидаемый ответ, “delay” – время ожидания перед запросом ответа, а “c” – флаг для ввода пользователем комментария, содержащий значение по умолчанию;
- Пользователь выбирает USB-порты своего компьютера, к которым подключены микроконтроллеры, на которые будут отправляться запросы и с которых будут считываться ответы;
- Пользователь запускает программу;
- Программа проверяет конфиг на корректность;
- Программа интерпретирует конфиг и, если необходимо, выдаёт диалоговые окна для общения с пользователем;

- По окончании тестирования выдаёт результаты и предлагает сохранить их в отдельный файл или в базу данных (подключенную заранее или выбранную уже после тестирования).

Для реализации нашего решения мы будем использовать Python как основной язык программирования; PyQt5, который имеет реализацию под Python, и Qt Designer для создания интерфейса; библиотеку pyserial для связи с микроконтроллерами; MongoDB и MySQL для базы данных; json и dhall [2] для форматирования конфигурационных файлов.

Для тестирования нашего решения была взята задача из курса “Интернет вещей”, где мы использовали два микроконтроллера Arduino и специально созданное для передачи по азбуке Морзе и демонстрации символов устройство MorseShield. К первому микроконтроллеру был подключен MorseShield. Полученная комбинация Arduino+MorseShield рассматривалась нами как единая печатная плата в рамках данной задачи. Второй микроконтроллер вёл мониторинг и принимал данные с MorseShield, которые записывались, как описано выше.

Выводы.

Мы использовали для валидации решения хорошо известный протокол передачи данных - азбуку Морзе. Так как мы знаем, как именно мы кодируем последовательность символов, это хороший способ сделать валидацию нашего решения. При работе комплекса были выявлены отклонения принимаемого сигнала от эталонного, что позволило ускорить отладку. В дальнейшем работа будет направлена на то, чтобы сделать программно-аппаратный комплекс более универсальным для работы с произвольной печатной платой.

Список использованных источников:

1. Робертс С, Рентюк В. Решения проблемы пульсаций и помех DC/DC-преобразователей: входная и выходная фильтрация. Компоненты и технологии. 2015(8):74-82.
2. Хабр [Электронный ресурс]: Зиновьев О. (@WellsBart) - Язык конфигураций Dhall как еще один способ написания манифестов для Kubernetes. // 2021. URL: <https://habr.com/ru/company/flant/blog/593321/>. (Дата обращения: 17.10.2022).