



УДК 004.415.2

DOI: 10.22184/NanoRus.2019.12.89.164.165

СИСТЕМА АВТОМАТИЗАЦИИ ИСПЫТАНИЙ БОРТОВОГО СВЧ-ОБОРУДОВАНИЯ

AUTOMATION SYSTEM OF ONBOARD MICROWAVE EQUIPMENT VERIFICATION

МАДУМАРОВ ТАЛГАТ АСХАТОВИЧ

madrook00@gmail.com

MADUMAROV TALGAT A.

madrook00@gmail.com

СТЕФАНЦОВ АЛЕКСЕЙ ВЯЧЕСЛАВОВИЧ

stef_zel@mail.ru

STEFANTSOV ALEXEY V.

stef_zel@mail.ru

АО «Научно-исследовательский институт «Субмикрон»
124460, г. Москва, г. Зеленоград,
Георгиевский просп., 5, стр. 2

Submicron Scientific Research Institute JSC
bld. 2, 5 Georgievsky Ave.,
Zelenograd, Moscow, 124460

Во время параллельной разработки СВЧ-аппаратуры и измерительного стенда для автоматизации испытаний возникают риски, связанные с поздней интеграцией. Построение стенда по клиент-серверной архитектуре и использование технологии REST позволяют разрабатывать и тестировать компоненты изолированно. Это уменьшает время интеграции и позволяет начинать разработку ПО и аппаратуры одновременно.

Ключевые слова: верификация оборудования; СВЧ; АФАР; REST; C++; системная интеграция.

Some risks associated with later integration occur due to parallel design of microwave hardware and verification testbench. Using client-server architecture and REST technology makes it possible to develop and test all components separately. This can reduce integration time and allows starting design of hardware and verification software simultaneously.

Keywords: hardware verification; microwave; AESA; REST; C++; system integration.

Во время проведения испытаний СВЧ-приборов в целом и АФАР в частности приходится проводить большое количество измерений, а затем обрабатывать полученные данные. Для автоматизации этих повторяющихся действия необходима интегрированная среда проведения испытаний, которая включала бы в себя следующий функционал (рис. 1):

- управление целевым устройством, изменение режимов работы устройства, получение телеметрии;
- управление источниками питания;
- управление источниками сигналов для измерения электрофизических испытаний;
- управление измерительными приборами, такими как векторный анализатор для измерения S-параметров и анализатор спектра;
- обработка полученной информации и сохранение результатов в формат, удобный как для обработки вручную, так и для

последующей обработки в специализированных программах, таких как Matlab.

Система автоматизации испытаний работает следующим образом:

- интегрированная среда проведения испытаний содержит текстовый редактор, интерфейс отображения состояния целевого устройства и протокол проведения испытания;
- в текстовом редакторе пользователь пишет программу испытаний, которая транслируется и исполняется интегрированной средой. Программа включает в себя логические операции, команды работы с измерительной аппаратурой, управления целевым устройством, а также команды обработки и сохранения результатов измерения;
- текстовые сообщения от объекта испытаний и от измерительных приборов отображаются в окне протокола испытаний; телеметрия с целевого устройства отображается в реальном времени;

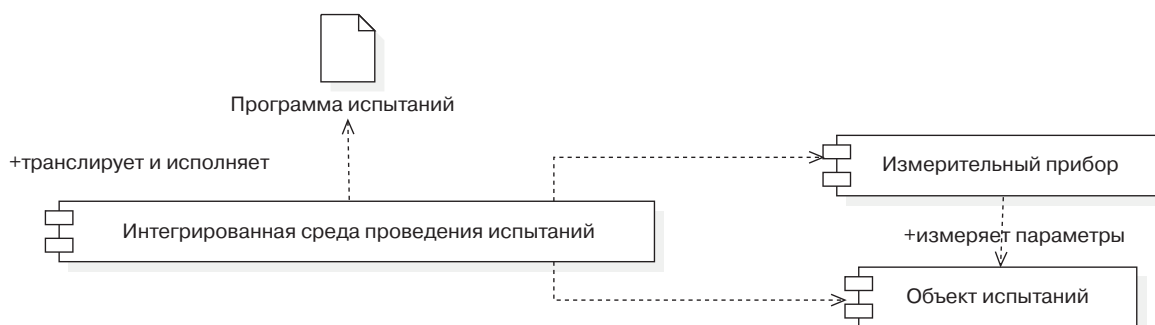


Рис. 1. Диаграмма компонентов испытательного стенда

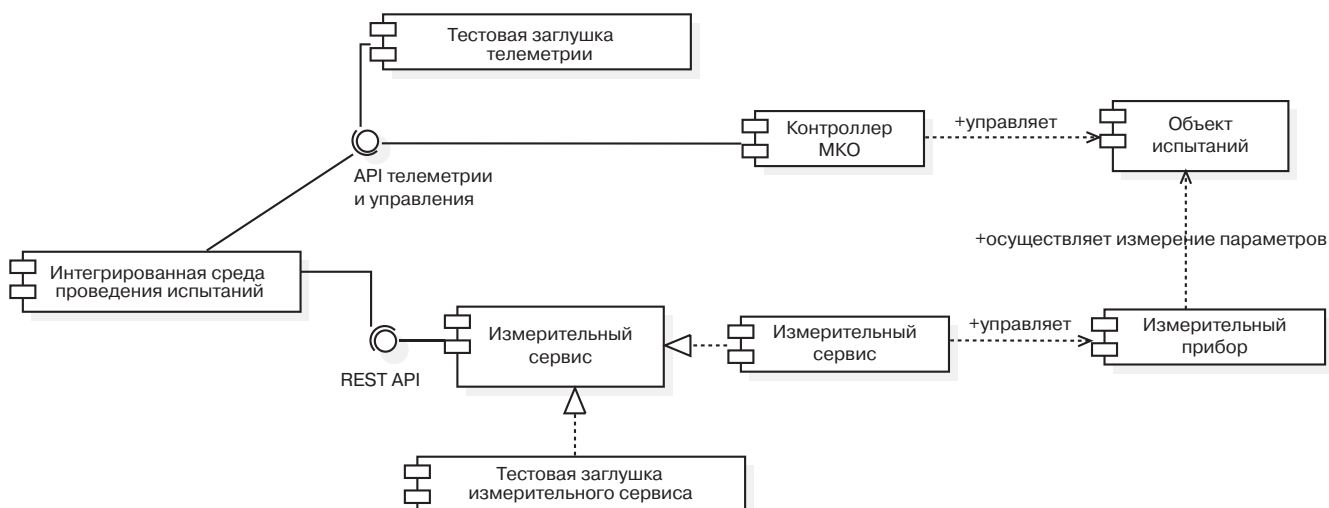


Рис. 2. Диаграмма компонентов системы

- результаты проведения испытаний пользователь может сохранить в виде отчета в удобном формате.

Во время разработки подобных систем возникают накладные расходы, связанные с рисками задержки разработки аппаратной части испытательного стенда. В связи с этим на разработчиков ПО ложится задача минимизировать эти риски.

Для снижения рисков при интеграции, а также для того, чтобы начать разработку ПО и аппаратной части стенда одновременно, было принято решение о создании тестового окружения, состоящего из заглушек измерительных сервисов и телеметрии. Поскольку сам прибор уже был разработан, необходимости в моделировании управления целевым устройством не возникло.

Для связи интегрированной среды проведения испытаний с измерительными приборами было принято решение использовать технологию REST (Representational State Transfer) [1]. Достоинством данной технологии является строгая формализация. В качестве формата данных был принят стандарт JSON [2]. Формализация протокола обмена данными на самых ранних этапах разработки ПО позволила в кратчайшие сроки создать на языке Python тестовый HTTP-сервер для отладки взаимодействия с измерительной аппаратурой. Также была создана модель телеметрии. Все это позволило выделить в архитектуре системы высокорисковые компоненты и четко оформить границы между компонентами с высокой и низкой степенью риска, что в свою очередь позволило создать тестовую интеграционную среду (рис. 2).

Было налажено активное и своевременное взаимодействие как с заказчиком, так и с контрагентами, что позволило оперативно уточнять требования и затем быстро вносить изменения в код. Поскольку протоколы обменов между компонентами


системы не менялись, все изменения легко вносились и сразу же тестировались. При этом тестовое окружение позволяло тут же проводить интеграционное тестирование взаимодействия между всеми компонентами, что повысило надежность разрабатываемого продукта.






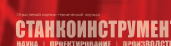
Для сохранения результатов было решено использовать формат CSV [3]. Данные в этом формате представлены в виде таблицы полей, разделенных запятыми. Такие данные можно легко обрабатывать как вручную при помощи редакторов таблиц, так и с помощью ЭВМ.

Использование технологии REST, автоматизированное модульное и интеграционное тестирование, формализация протоколов на ранних этапах, а также непосредственная оперативная связь с заказчиком и контрагентами позволили уже на ранних этапах создать тестовое окружение. В связи с этим стало возможно вести интеграцию сложной системы непрерывно. В итоге после разработки испытательного стенда окончательная интеграция ПО с аппаратурой продолжалась всего две недели. Во время разработки данной системы был приобретен ценный опыт по параллельной разработке программной и аппаратной частей испытательных стендов, который будет применим в будущих разработках.

ЛИТЕРАТУРА

1. Rodriguez A. *Restful web services: The basics* // IBM developerWorks, 2008. Vol. 33.
2. Aziz A., Mitchell S. *An Introduction to JavaScript Object Notation (JSON) in JavaScript and .NET* // Microsoft Developer Network, 2007.
3. <https://ru.wikipedia.org/wiki/CSV>.


ТЕХНОСФЕРА
 РЕКЛАМНО-ИЗДАТЕЛЬСКИЙ ЦЕНТР
www.technosphere.ru

Цифровая экономика