

Ende-zu-Ende-Tests für IoT-Anwendungen



Zu jeder qualitativ hochwertigen Wireless IoT-Anwendung mit Embedded Systemen gehört am Anfang des Lebenszyklus eine umfassende Ende-zu-Ende-Feldtestphase und während der gesamten Nutzungsphase ein permanentes Monitoring mit geeigneten Telemetriedaten. Dafür sind geeignete Methoden und Werkzeuge erforderlich, die auch innerhalb der Anwendungsumgebung einsetzbar sind.

Embedded Systems-basierte Wireless IoT-Anwendungen sind komplexe technische Systeme. Im Vorfeld eines erfolgreichen Praxis-einsatzes müssen daher umfangreiche Tests stattfinden, die weit über die Erfüllung der Compliance-Pflichten hinausgehen. Wichtig ist, mit Hilfe geeigneter Testmethoden möglichst viele Schwachstellen und Fehler zu finden und zu beseitigen, um die erforderliche Zuverlässigkeit und Sicherheit zu gewährleisten. Während der gesamten Nutzungsphase durch den Anwender sollten darüber hinaus ein permanentes Monitoring mit geeigneten Telemetriedaten für DevOps-Prozesse und Over-the-Air (OTA)-Software-Updates für alle Systembausteine zur Anwendung kommen.

Auswahl der Testmethoden

Den Entwicklern von IoT-Komponenten und Anwendungen stehen inzwischen zahlreiche Testmethoden zur Verfügung. Die Auswahl des

passenden Methodenmix hängt von verschiedenen Faktoren ab. Neben dem spezifischen Know-how innerhalb des Entwicklerteams haben die jeweils zu testenden Komponenten, IoT-Systemanforderungen und Compliance-Aspekte einen großen Einfluss auf den „Was wird wie getestet?“-Fragenkomplex. Aber auch Terminpläne und Budgets hinterlassen ihre Spuren. Zu den Kernkompetenzen eines IoT-Entwicklerteams und somit zum Pflichtprogramm der durchzuführenden Tests sollten allerdings auf jeden Fall die folgenden vier Methoden gehören:

Unit-Test

Mit diesem Softwaretest werden z. B. die Module und Funktionen einer IoT-Baugruppen-Firmware isoliert getestet, um die fehlerfreie Funktion sicherzustellen. Ein Beispiel für einen Unit-Test wäre die Überprüfung einer IoT-Sensor-Codefunktion hinsichtlich der korrekten Umwandlung einer analogen Eingangsmessgröße in einen digitalen Messwert.

Integrationstest

Dabei wird die Embedded Systems-Hard- und Software mehrerer IoT-Komponenten gemeinsam getestet, um die einwandfreie Zusammenarbeit sicherzustellen. Ein Beispiel ist der MQTT-Kommunikationstest zwischen Sensor und

Gateway, um sicherzustellen, dass die Sensormesswerte korrekt übertragen und empfangen werden und auf dem Gateway die exakte Messgröße vorliegt.

Funktionstest

Dieser Test dient der Validierung spezifischer IoT-Baugruppenfunktionen hinsichtlich ihrer korrekten Funktion. Ein Anwendungsbeispiel wäre der Test eines KI-basierten Machine Vision-Sensors, dem verschiedene physische Objekte mit bestimmten Merkmalen präsentiert werden. Dabei wird z. B. das Datenpaket am Sensorausgang hinsichtlich des Klassifizierungsergebnisses geprüft.

Systemtest

Mit Hilfe einer solchen Testmethode wird ein gesamtes System als Einheit geprüft, um festzustellen, ob die erforderliche Systemfunktionalität gegeben ist. Ein Beispiel wäre der vollständige Test eines IoT-Systems mit allen Sensoren, Aktoren, Infrastrukturbaugruppen, Netzwerkverbindungen sowie der Cloud-basierten Anwendungssoftware. Da ein solcher Ende-zu-Ende-Test in der Regel unter Laborbedingungen stattfindet, werden einige Teilfunktionen (z. B. die Benutzerinteraktionen oder das Sensordatenbild mit den realen Umgebungsbedingungen) mit geeigneten Hilfsmitteln simuliert.

Systemtest reicht nicht

Eine weitere sehr wichtige Testmethode ist der Ende-zu-Ende (E2E)-Test. Er ähnelt auf den ersten Blick zwar dem Systemtest, liefert aber in Praxis tiefere Einblicke hinsichtlich möglicher Schwachstellen und Fehler. Bei einem E2E-Test wird ebenfalls das gesamte System überprüft, ausgehend von den Embedded Systemen der IoT-Endpunkte (Sensoren, Aktoren) über die Infrastrukturkomponenten bis zur Benutzeranwendung. Dabei werden neben den Kommunikationsschichten (Netzwerke, Protokolle, Cloud-Services usw.) aber auch die Benutzerschnittstellen einbezogen. Der E2E-Test deckt somit den vollständigen Daten- bzw. Informationsfluss ab, also von der Datenerfas-

Autor:

Klaus-Dieter Walter
CEO

SSV Software Systems GmbH
www.ssv-embedded.de

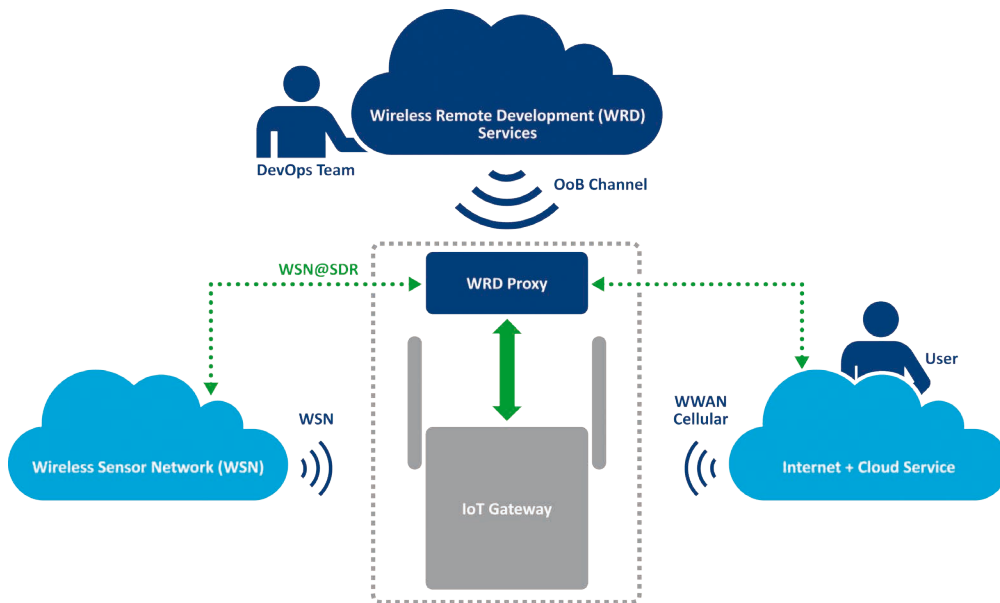


Bild 1: Ein effektiver Ende-zu-Ende-Test für Wireless IoT-Lösungen bezieht zum einen die Benutzeraktivitäten ein und wird zum anderen in Zielumgebungen mit realistischen Umgebungsbedingungen durchgeführt.
 © Wikipedia (https://de.wikipedia.org/wiki/Software_Defined_Radio)

sung mittels der IoT-Sensorik über die Datenübertragung und -verarbeitung in der Cloud bis hin zur Darstellung der Daten in einer Benutzeroberfläche.

Benutzerinteraktion im Blick

Im Unterschied zum Systemtest werden die Benutzerinteraktion berücksichtigt, d. h. wie beispielsweise der Endbenutzer mit dem System interagiert und ob das gesamte System auch in diesem Kontext wunschgemäß funktioniert. Ein weiterer Unterschied zum Systemtest ist, dass ein solcher E2E-Test nicht im Labor, sondern so realitätsnah wie möglich mit echten Umgebungs- und Netzwerkbedingungen sowie realistischen Benutzerinteraktionen direkt in einer typischen Anwendungsumgebung erfolgen sollte. Ein effektiver IoT-Ende-zu-Ende-Test ist daher ein Feldtest, beispielsweise mit ausgesuchten Pilotkunden über einen längeren Zeitraum. Dafür sind Entwicklerwerkzeuge erforderlich, die sich per Fernzugriff bedienen lassen, aber unabhängig von den IoT-Kommunikationsverbindungen der zu testenden Anwendungen funktionieren (also ein „Wireless Remote Developing“ ermöglichen).

Vorteile der Remote Tests

Solche Remote Tests liefern erfahrungsgemäß weitreichende

Erkenntnisse, die sich mit einem Systemtest in einer Labor- bzw. Entwicklungsumgebung in der Regel nicht gewinnen lassen. Ein Beispiel ist immer wieder die unvorhersehbare Kreativität der Anwendungsnutzer und die dadurch verursachten Interaktionssequenzen. Aber auch die Nutzung von Mobilfunk- oder lizenzfreien LPWA-Netzen mit IoT-Gateways an unterschiedlichen Standorten (sowohl national als auch international) bietet diverse Überraschungen. In der Praxis zeigt sich diesbezüglich immer wieder, dass IoT-Anwendungen beispielsweise dynamische Veränderungen in Mobilfunknetzen (z. B. Verfüg-

barkeitseinschränkungen, Änderungen einer LoRaWAN-Infrastruktur, Latenzzeitunterschiede, einen 4G-auf-2G-Fall-back oder die Besonderheiten einer Global-SIM-Karte) nur unzureichend berücksichtigen und dadurch Probleme im 24/7-Dauerbetrieb auftreten.

Transparente Funkschnittstellen

Ein sehr wertvolles Werkzeug für E2E-Tests mit Wireless IoT-Anwendungen sind Software Defined Radio (SDR)-Systeme (Bild 1). Eine wesentliche SDR-Eigenschaft ist, dass die relevanten Parameter eines Funksystems, wie Frequenzbereich,

Modulation, Bandbreite, Zeitverhalten, Kanalkodierungsverfahren usw. durch alleinige Änderungen der Software bzw. der entsprechenden Parametervariablen erreicht werden. Dadurch lässt sich ein SDR beispielsweise über die passenden Parameterwerte und Softwarefunktionen an die Funktechnik einer IoT-Anwendung anpassen. Anschließend ist ein passives Mithören sowie die Endpunkt-Emulation von verschiedenen Funkprotokollen in unterschiedlichen Frequenzbereichen möglich (beispielsweise LoRa, Bluetooth, LTE). Mit einem SDR lassen sich auch verschiedene Cyberangriffsszenarien durchspielen (z. B. Wireless Replay Attacks), um das Verhalten des zu testenden Systems diesbezüglich zu untersuchen.

Vielseitiges Werkzeug

SDRs sind daher ein äußerst vielseitiges Werkzeug für die Funkkommunikationsprüfung in Wireless IoT-E2E-Tests. Sie ermöglichen flexible, Budget-freundliche und weitreichende RF-Tests, um sicherzustellen, dass die drahtlose Kommunikation in einem IoT-System robust, sicher und zuverlässig im Rahmen der jeweiligen Anforderungen auch im Praxiseinsatz funktioniert.

E2E-Tests sind Feldtests

Im Rahmen des IoT-E2E-Tests innerhalb einer echten Anwendungsumgebung sollte möglichst eine kontextbezogene Fragenliste abgearbeitet werden. Die Tabelle 1 liefert hierzu ein Beispiel. Als Umfeld dient die Wireless-IoT-Anwendung aus der Bild 1.

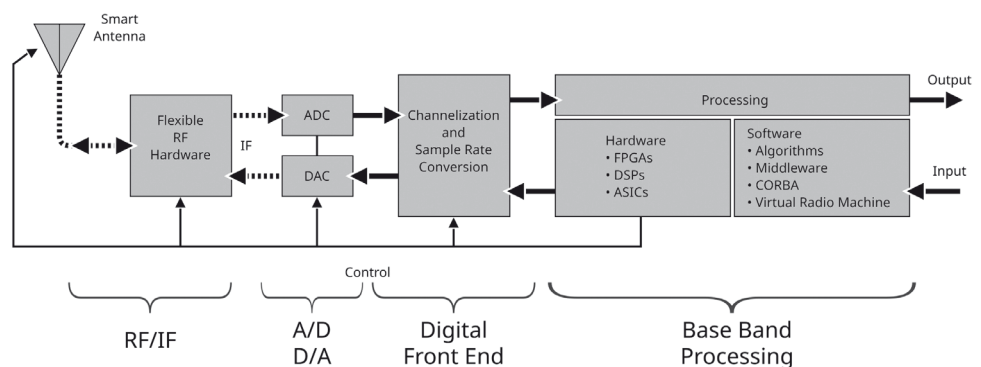


Bild 2: Die Hardware eines Software Defined Radio (SDR) besteht typischerweise aus einem (analogen) Sender- und Empfängermodul als RF-Interface, sowie A/D- und D/A-Wandlern plus einem digitalen Base Band Processing, also z. B. eine Software-basierte digitale Signalverarbeitung mit Digitalen Signalprozessoren (DSPs).

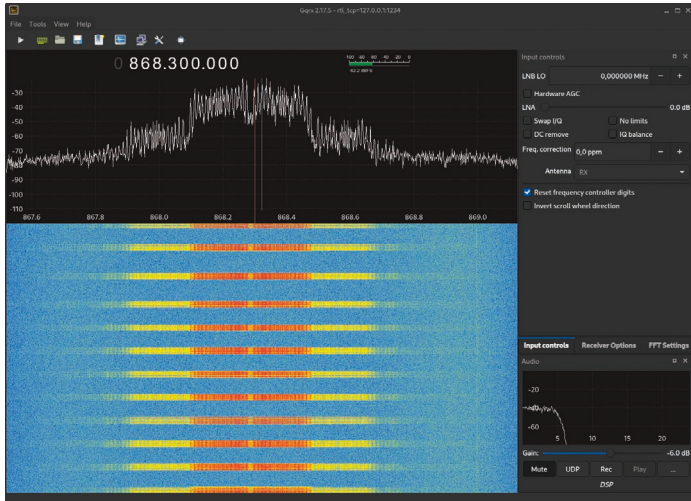


Bild 3: Mit einem SDR und der passenden Software lässt sich z. B. der FFT-Plot mit Wasserfalldiagramm für die Datenübertragung eines Wireless-IoT-Sensor-Netzwerks im 868 MHz-Frequenzbereich visualisieren.

Besonders wichtig für den E2E-Test ist in diesem Kontext, dass der vollständige Datenfluss vom IoT-Sensor als Datenquelle über das Wireless Sensor-Netzwerk und IoT Gateway sowie den Cloud Service bis zur Daten-konsumierenden App eines Benutzers gänzlich und so transparent wie möglich ausgetestet wird. Dabei sind selbstver-

ständig echte Nutzeraktivitäten zu berücksichtigen.

Dauertest

Nachdem das erste neu entwickelte System an einem Standort den E2E-Test erfolgreich durchlaufen hat, sollte man das gesamte Testscenario sinnvollerweise als Dauertest auf weitere Systeme

an verschiedenen Orten ausdehnen. Die zusätzlichen Systemstandorte werden dabei hinsichtlich der Umgebungsbedingungen möglichst nach Worst-Case-Szenarien ausgesucht. Für die einzelnen Testabläufe ist dann ein gewisser Automatisierungsgrad erforderlich, um Langzeittestphasen über mehrere Tage bzw. Wochen mit akzeptablem Aufwand zu ermöglichen. Im Rahmen solcher Testautomatisierungsaufgaben kommen in der Regel verschiedene Scriptsprachen, wie z. B. Python, zum Einsatz.

Anwendungsbeispiel

Die Wireless Sensor-Netzwerk (WSN)-Verbindungen in Bild 1 und die darin enthaltenen Embedded Systeme sind verschiedenen Störungen ausgesetzt, die von Standort zu Standort teilweise erheblich variieren. Typische Beispiele sind RF-Interferenzen und Batteriebetriebbedingte Reichweiteneinschränkungen, die unterschiedliche Kommunikationsprobleme verursachen können. Dazu gehören Interferenzen, Datenpaketverluste (RF Packet Loss), unterschiedliche Latenzzeiten eines Kommunikationskanals

usw. Insofern ist es im Rahmen des E2E-Tests sinnvoll, den jeweiligen Packet Loss bzw. Datendurchsatz für eine bestimmte Umgebung zu ermitteln. Dafür sollten bereits im Rahmen der Entwicklung entsprechende Funktionen (z. B. ein Wireless-Ping) im IoT-Sensor und Gateway vorgesehen werden. Auch eine Embedded-SDR-Funktion ist hilfreich, um während der gesamten Nutzungsphase ein permanentes Monitoring mit geeigneten WSN-Telemetriedaten für DevOps-Konzepte zu liefern.

Acht wichtige Fragestellungen

Im Rahmen eines Wireless IoT-Ende-zu-Ende (E2E)-Tests lassen sich beispielsweise acht einzelne Fragestellungen bearbeiten. Ein wichtiges Werkzeug zur Beantwortung der einzelnen Fragen ist ein Software Defined Radio (SDR). Zusammen mit einer geeigneten Software, wie z. B. GNU Radio (<https://www.gnuradio.org/>), lässt sich ein SDR als Funkempfänger zum passiven Mithören (Sniffer Mode) und als aktiver WSN-Netzwerkteilnehmer (Sendemodus zur Endpunkt-Emulation) nutzen. ◀

	Fragestellung	Beschreibung
1.	Hat der WSN-Knoten etwas gesendet?	Die einzelnen Funksensoren im WSN senden in einer sternförmigen Netzwerktopologie diverse Datenpakete an das Gateway. Mit Hilfe des SDR Sniffer Mode lässt sich feststellen, was jeweils gesendet wurde.
2.	Hat das Gateway es gehört?	Durch einen Gateway Service Link-Zugriff wird geprüft, ob die zu 1. gehörenden WSN-Daten auch empfangen wurden. In der Regel lösen die Daten aus dem WSN eine Datenübertragung vom Gateway an den Cloud Service aus.
3.	Hat der Cloud Service die Sendedaten erhalten?	Mit Hilfe gezielter Cloud Service-Zugriffe lässt sich feststellen, ob die vom IoT Gateway per WWAN gesendeten Daten den Cloud Service erreicht haben.
4.	Konnte der Cloud Service die Sendedaten verwerten (also z. B. dekodieren/zuordnen)?	Die an den Cloud Service gesendeten Daten müssen bestimmte Vorgaben (beispielsweise bzgl. der Zugriffsrechte, Datenformate usw.) erfüllen, damit sie bestimmungsgemäß verarbeitet werden können. Per Cloud Service-Zugriff lässt sich das Ergebnis prüfen.
5.	Hätte der Cloud Service etwas zurücksenden sollen bzw. müssen?	Über Cloud Service-Zugriffe lässt sich feststellen, ob die Cloud etwas an das IoT Gateway sendet oder nicht (hier sind evtl. Steuerungsfunktionen der Cloud bzw. Benutzerinteraktionen einzubeziehen, z. B. mittels einer externen Smartphone-App).
6.	Hat der Cloud Service etwas an das Gateway gesendet?	Per Gateway Service Link-Zugriff lässt sich prüfen, ob eine unter 5. identifizierte WWAN-Datenübertragung vom Gateway auch empfangen und verarbeitet wurde.
7.	Hat das Gateway diese Daten an den WSN-Knoten gesendet?	Falls die WWAN-Datenübertragung vom Cloud Service zum Gateway eine WSN-Datenübertragung verursacht, lässt sich per SDR Sniffer Mode prüfen, ob die WSN-Daten vom Gateway erwartungsgemäß verschickt wurden.
8.	Hat der WSN-Knoten die Daten empfangen?	Diese Frage lässt sich nur sicher beantworten, wenn das zum Einsatz kommende Kommunikationsprotokoll eine Bestätigung vorsieht, die sich mit Hilfe des SDR Sniffer Mode aufzeichnen lässt oder wenn eine zusätzliche Debug-und-Test-Verbindung zum WSN-Knoten besteht (z. B. per SWD).

Tabelle 1: Fragen im Rahmen eines Wireless-IoT-Ende-zu-Ende (E2E)-Tests