HomeMatic® goes Android

HomeMatic®-Status und -Bedienung per App

Dank öffentlich dokumentierter Schnittstellen, Bibliotheken und Entwicklungsumgebungen ist es dem ambitionierten HomeMatic-Nutzer heute möglich, eine Android-App für die Steuerung der eigenen HomeMatic-Haustechnik nach individuellem Bedarf zu entwickeln. Wir zeigen anhand eines Beispiels, wie man hier ansetzt und sich sowohl Ereignisse melden lässt als auch die Bedienung von HomeMatic-Geräten via App realisieren kann.

Alles da …

Die Ansteuerung und Integration von HomeMatic-Geräten lässt sich anhand der öffentlich dokumentierten XML-RPC-Schnittstelle [1] sowohl für Funk- als auch für drahtgebundene Komponenten einfach vornehmen. Neben entsprechenden Aktoren und/oder Sensoren wird hierfür eine HomeMatic-Zentrale benötigt, wobei es egal ist, ob es sich um eine CCU1-Zentrale oder eine CCU2 handelt.

Da Bibliotheken zur Nutzung von XML-RPC für die meisten Plattformen und Programmiersprachen in hoher Qualität zur Verfügung stehen, lassen sich beispielsweise für Betriebssysteme von stationären Computern (z. B. Windows, Mac OS X, Linux) oder für mobile Geräte (z. B. Android, iOS) eigene Lösungen realisieren. Google bietet mit Android ein Betriebssystem für Smartphones und Tablets, welches über ein öffentliches und kostenlos zu beziehendes SDK[1] [2] verfügt und eine Ausführung eigener Apps ohne zusätzlich zu zahlende Entgelte oder Registrierungen bei einem Hersteller erlaubt.

Im folgenden Artikel soll gezeigt werden, wie sich ein Auslesen des Status und die Bedienung von HomeMatic-Geräten mittels einer eigens entwickelten Android-App realisieren lässt.

Die XML-RPC-Schnittstelle der HomeMatic®-Zentralen

Die Kommunikation z. B. über das Netzwerk durch den Aufruf von Methoden bzw. Funktionen einer Schnittstelle eines Geräts bzw. Systems wird Remote Procedure Call (RPC) genannt, hierbei wird für die Schnittstelle ein RPC-Server genutzt.

Die Kommunikation zwischen der Anwendung und dem Server kann in unterschiedlichen Formaten erfolgen, für die HomeMatic-Zentralen wird XML als Format verwendet.

Ein Beispiel für einen derartigen Methodenaufruf der XML-RPC-Schnittstelle ist das Abrufen einer Liste aller aktuell angelernten Geräte.

Die HomeMatic-Zentralen stellen für Funkkomponenten (BidCoS®-RF) als auch für drahtgebundene Geräte (BidCoS®-Wired) jeweils eine XML-RPC-Schnittstelle auf einem eigenen Netzwerk-Port zur Verfügung.

Die komplette öffentliche Spezifikation mit genaueren Informationen zum Zugriff auf die Schnittstelle ist auf der HomeMatic-Website [1] zu finden.

Für eine vereinfachte Demonstration der Nutzung der Schnittstelle nutzt die Android-App nur die in Tabelle 1 vorgestellten Methoden der HomeMatic-Zentralen.

Entwicklung einer eigenen Android-App

Die Entwicklung von Android-Apps erfolgt mit der Programmiersprache Java und der Entwicklungsumgebung Eclipse. Zusätzlich wird das Android-SDK (Download über [2]) für die Entwicklung benötigt.

Zum Testen und Nutzen der erstellten App kann ein über USB angeschlossenes Android-Gerät oder der im SDK enthaltende Android-Emulator genutzt werden. Grundlegende Kenntnisse in der objektorientierten Programmierung in Java sind hilfreich.

Eine Android-App besteht aus mehreren sogenannten Activities. Eine einzelne Activity repräsentiert ein Bedieninterface für einen einzelnen Bildschirm. Wie bei einer Desktopanwendung können auf dem Bedieninterface einzelne Elemente wie Buttons, Eingabefelder oder Textfelder platziert und entsprechend ausgewertet werden. Das Layout dieser Elemente wird in einer separaten XML-Datei definiert und mit der dazugehörigen Java-Klasse zur jeweiligen Activity verknüpft. Activities können sich gegenseitig mittels sogenannter Intents aufrufen, beispielsweise kann eine Activity, die die Telefonkontakte geordnet darstellt, eine Activity aufrufen, die das Schreiben einer Kurznachricht an einen zuvor ausgewählten Kontakt erlaubt.

Eine genaue und bebilderte Einführung in die Entwicklung einer „Hello World"-App (startbare Anwendung, die nur einen Text ausgeben soll) ist unter [3] zu finden.

Aufbau und Nutzung der HomeMatic®-App

Die erstellte Android-App soll als einfaches Beispiel zeigen, wie sich HomeMatic-Zentralen und deren angelernte Aktoren und Sensoren mittels XML-RPC auf einem Smartphone steuern lassen können. Für diesen Zweck nutzt die App hauptsächlich 2 Activities. Die eine, um eine Verbindung zu einer angegebenen Zentrale herzustellen und Informationen zu ihren aktuell angelernten Geräten zu erhalten, die zweite Activity, um den Status der Geräte auszulesen und zu bedienen.

Beim Start der App wird die in Bild 1 abgebildete Activity dem Nutzer gezeigt. Es müssen die Adresse

Bild 1: Der Startbildschirm der App: Setzen der Verbindungseinstellungen zum Auslesen der angelernten Aktoren und Sensoren

(Hostname oder IP-Adresse) der Zentrale und ein Port für die Verbindung zum gewünschten XML-RPC-Server angegeben werden. Für den Zugriff auf die Zentrale muss sich das jeweilige Smartphone bzw. Tablet im selben logischen Netzwerk befinden, also z. B. per Kabel und/oder WLAN am selbem Router angeschlossen sein. Um die Erreichbarkeit zu testen, kann versucht werden, mit dem Browser des Geräts auf die Web-Oberfläche der Zentrale zuzugreifen. Eine Erreichbarkeit über das Internet ist nicht gegeben, da die verwendeten Schnittstellen für lokale Netzwerke ausgelegt sind.

Mit der Schnittstelle für Funkkomponenten (BidCoS®-RF) kann über Port 2001, mit der für drahtgebundene Geräte (Wired) über Port 2000 kommuniziert werden (siehe öffentliche Spezifikation, [1]). Der jeweilige Port muss auf der Oberfläche eingetragen werden.

Nachdem eine Verbindung zu einer HomeMatic-Zentrale aufgebaut wurde, werden die angelernten Geräte ausgelesen. Dies geschieht mittels der XML-RPC-Methode „listDevices", die virtuelle Fernbedienung „RCV50" wird zwecks Übersichtlichkeit bei dem Aufbau der Geräteliste ignoriert.

Für alle relevanten Kanäle zur Bedienung und/oder zum Auslesen des Zustands der zuvor erhaltenen Geräteliste werden die entsprechenden Parameterbeschreibungen mithilfe der Methode „getParamsetDescription"

Tabelle 1

Genutzte XML-RPC-Methoden der HomeMatic®-Zentrale

Methodenname	Parameter	Rückgabewert	Beschreibung
listDevices	[ohne]	Array<DeviceDescription>	Es wird eine Liste aller aktuell angelernten Geräte in Form einer DeviceDescription-Struktur zurückgegeben. Die Struktur enthält eine Vielzahl von Daten zu dem Gerät, beispielsweise den Typ oder seine Adresse.
getParamsetDescription	String address, String paramset_type	ParamsetDescription	Liest die verfügbaren Parameter eines Gerätekanals aus. Der zweite Parameter ist hier vereinfacht immer „VALUES". Die Antwort ist eine Liste aus einzelnen Parameterbeschreibungen.
getValue	String address, String value_key	ValueType	Liest den Wert eines Parameters eines Gerätekanals aus. Beispiele für „value_key" sind „LEVEL" (u. a. Dimmer) oder „STATE" (Schaltstatus, an/aus).
setValue	String address, String value_key, ValueType value	[ohne]	Setzt den Wert eines Parameters eines Gerätekanals. Ein Schaltaktor kann beispielsweise durch den value_key „STATE" und value „false" ausgeschaltet werden.
init	String url, String interface_id	[ohne]	Meldet sich als eigene Logikschicht bei der XML-RPC-Schnittstelle an. Der Parameter „url" entspricht der URL, unter welcher die App erreicht werden kann. Die „interface_id" dient zur Erkennung von erhaltenden Events der Zentrale und ob sie für die App relevant sind.

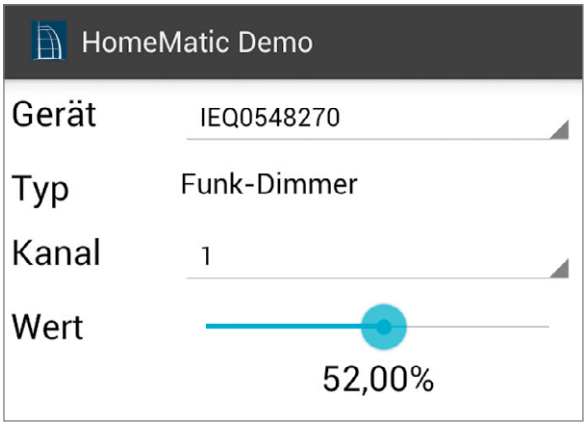

Bild 2: Steuerung eines Funk-Dimmers mit der erstellten App

abgerufen. Wenn dieser Vorgang erfolgreich war, wird der Benutzer auf die Activity zur Nutzung der ausgelesenen Informationen weitergeleitet (siehe Bild 2).

Sie enthält eine Liste mit den Seriennummern der zuvor erhaltenen Geräte, aus der das gewünschte Gerät ausgewählt werden kann, anschließend wird der gewünschte Aktor- (Bedienen) bzw. Sensorkanal (Auslesen von Messwerten) ausgewählt.

Sofern für den jeweiligen Gerätetyp eine Beschreibung in der App hinterlegt ist, wird diese unterhalb der Geräteauswahl angezeigt, um eine Erkennung des jeweiligen Geräts zu erleichtern. Beispielsweise entspricht der Typ „HM-LC-Dim1PWM-CV" der Beschreibung „Funk-Dimmer".

Danach kann der zu nutzende Kanal des ausgewählten Geräts bestimmt werden. Der Wartungskanal „0" aller Geräte wird nicht zur Nutzung in der App angeboten. Wenn der gewählte Kanal einen Parameter besitzt, der bedient werden kann oder über den Werte abgerufen werden können, wird eine entsprechende Oberfläche zur Nutzung des Kanals angezeigt. Der erste Kanal des zuvor genannten Dimmers besitzt z. B. den Parameter „LEVEL", der dem aktuellen Dimmwert bzw. -level entspricht.

Innerhalb der Parameterbeschreibung gibt der Schlüssel „TYPE" Aufschluss darüber, ob der Wert des Parameters als Gleitkommazahl ausgedrückt wird. Für die Typen „FLOAT" (Gleitkommazahl), „INTEGER" (Ganzzahl) und „BOOL" (boolescher Wert) wird eine passende Bedienoberfläche zur Nutzung des Geräts angeboten. Der Schlüssel „WRITEABLE" zeigt, ob der Parameter schreibbar ist, womit erkenntlich wird, ob es sich bei dem Kanal des Geräts um einen Aktor oder einen Sensor handelt. Auch werden auf diese Weise Informationen über das Minimum („MIN") und Maximum („MAX") des Parameterwerts übermittelt.

Wenn der Wert des jeweiligen Kanals noch nicht abgerufen wurde, wird mittels der XML-RPC-Methode „getValue" (siehe Tabelle 1) der aktuelle Zustand des Kanals abgerufen. Beim Verändern der jeweiligen Bedienoberfläche wird mittels „setValue" der aktuell eingestellte Wert geschaltet, sofern dieser schreibbar ist. In Bild 2 ist das Schalten des ersten Kanals eines Dimmers abgebildet.

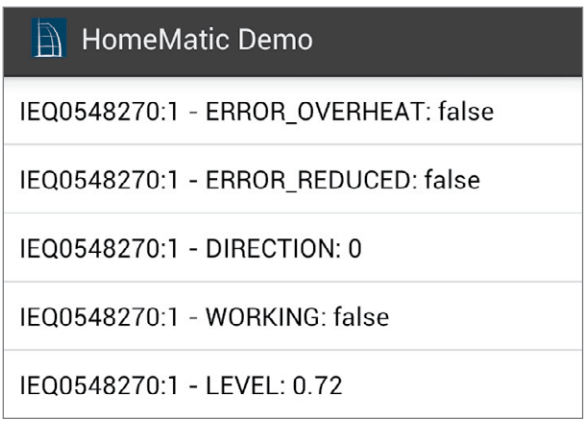

Bild 3: Empfangene Events nach Registrierung der App als eigene Logikschicht

Benachrichtigung von Zustandsänderungen durch Events

Damit die App über sich ändernde Gerätezustände, beispielsweise nach dem Schalten eines Geräts oder der zyklischen Sendung neuer Messwerte, mittels Events (Ereignisse) benachrichtigt wird, muss sie selbst einen XML-RPC-Server mit den in der Schnittstellenbeschreibung beschriebenen Methoden starten und diesen als sogenannte Logikschicht bei der Schnittstelle der Zentrale registrieren. Dies erfolgt durch die XML-RPC-Methode „init" (siehe Tabelle 1 für weitere Informationen).

In der App wird dieser Schritt beim Erstellen der Activity zum Verwalten der ausgelesenen Geräte durchgeführt. Während der Nutzung werden ankommende relevante Events entsprechend in der Oberfläche übernommen. Eine Liste der zuletzt empfangenen Events kann im Menü der Activity zur Gerätenutzung aufgerufen werden. Anhand des zuvor genutzten Dimmers würde die App nach dem Verändern des Werts ein Event von der verbundenen Zentrale erhalten, die empfangenen Events des Dimmers sind in Bild 3 dargestellt.

Download und Installation

Die beschriebene App kann unter [4] als Installationspaket (APK) inklusive aller Quelltexte kostenfrei heruntergeladen werden. Sie ist ab Android 4.0.3 und höher kompatibel.

Um die App installieren zu können, muss in den Einstellungen des Betriebssystems die Installation aus „Unbekannten Quellen" aktiviert werden (ab Android 4.0: Einstellungen – Sicherheit, unter 4.0: Einstellungen – Anwendungen).

Anschließend kann die App auf das per USB am Computer angeschlossene Smartphone oder Tablet übertragen werden. Nach der Übertragung wird die App über den Dateibrowser des Geräts installiert und kann gestartet werden. ELV

[1] *„Software Development Kit": Sammlung von Werkzeugen und Anwendungen, um eine Software zu erstellen, meist inklusive Dokumentation (siehe [5])*

Weitere Infos:

[1] Spezifikation der HomeMatic-XML-RPC-Schnittstelle: www.eq-3.de/software.html oder www.elv.de: Webcode #1263

[2] Android-SDK für Windows, Mac OS X und Linux: developer.android.com/sdk/index.html

[3] Einrichtung von Eclipse und bebilderte Einführung in die Erstellung einer „Hello World"-App: www.androidpit.de/de/android/wiki/view/Android_Anfänger_Workshop

[4] Download der erstellten HomeMatic-App: www.elv.de: Webcode #1264

[5] Erläuterung zum SDK: de.wikipedia.org/wiki/Software_Development_Kit