voco Call Button / HTTP – Systembeschreibung

Detlef Reil, 3. April 2012, zum voco Call Button, Version 110519

Software System

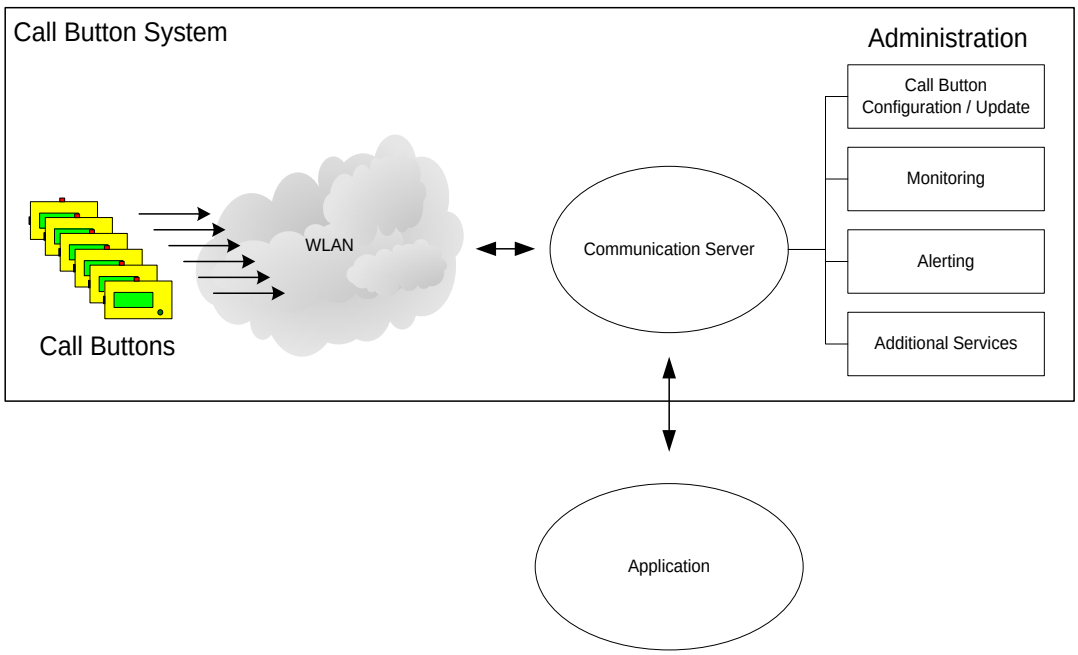
Für die Kommunikation zwischen den voco Call Buttons und der Applikation gibt es zwei Konzepte. Im ersten läuft sämtliche Kommunikation der Call Buttons über Rabus, das die Ereignisse an die Applikation vermittelt. Im zweiten Konzept werden die Ereignisse durch voco über HTTP direkt an die Applikation gesendet und Rabus dient nur zur Administration.

A - Rabus Kommunikations- und Administrationsserver

Sämtliche Kommunikation der voco Call Buttons läuft über *Rabus Server*, der die Ereignisse registriert und an die Applikation weitermeldet. Die Applikation kommuniziert mit *Rabus Server* über ein proprietäres Protokoll (Rabus-Protokoll). Die voco Call Buttons sind für die Applikation virtuell ständig erreichbar - Zustandsänderungen werden in Rabus gepuffert und während der regelmäßigen Weckereignisse der vocos an die Geräte weitergeleitet.

Der Vorteil dieser Lösung besteht darin, dass Rabus jeden Kommunikationsvorgang überwachen kann. Dadurch kann Rabus Störungen zuverlässig melden und ein weitgehendes Reporting vornehmen. Bei der HTTP-Kommunikation gehen die Ereignisse an Rabus vorbei, nur gelegentliche Administrationstransaktionen laufen dort auf. Ein vollständiges Logging ist so nicht möglich.

Rabus ist zurzeit nur als Windows-Software verfügbar.



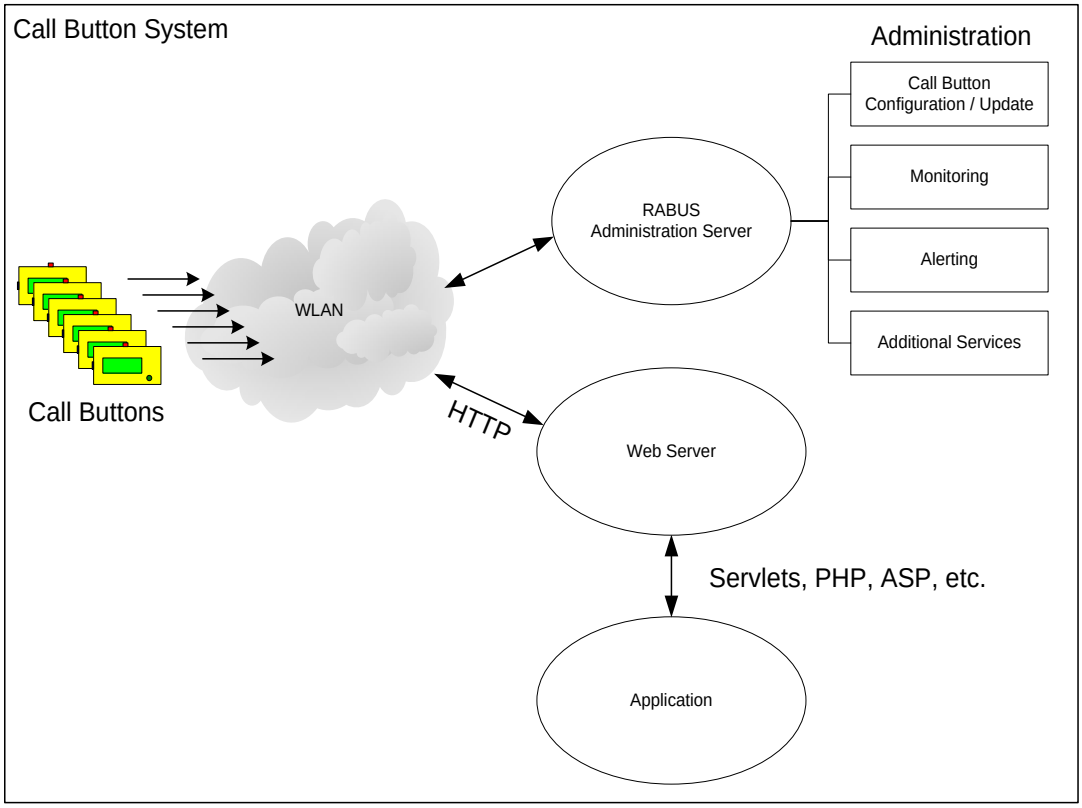
B – HTTP-Kommunikation über Webserver

Ein voco Call Button-Ereignis (z.B. Knopfdruck) baut eine direkte HTTP-Verbindung zum Webserver auf und fordert über die konfigurierbare URL eine Response an, die in den Header-Optionen Informationen über den darzustellenden Displayinhalt, LED-Stati und eine optionale Weckzeit enthält (s.u.). Anschließend wird die Verbindung wieder abgebaut.

Rabus als Administrationsserver

Zur Überwachung und Administration der Call Buttons kann Rabus verwendet werden. Die Funktion als Kommunikationsserver kommt in diesem Fall nicht zum Einsatz.

In regelmäßigen Abständen bauen die Call Buttons zur Überwachung eine Verbindung zum Rabus Administrationsserver auf. Rabus kann dann Konfigurationsänderungen oder Firmware-Updates übermitteln und einen Alarm beim Administrator auslösen, wenn das System nicht ordnungsgemäß funktioniert (Batterie leer, Call Button meldet sich nicht mehr,…)



Ist der *Rabus Server* nicht vorhanden oder nicht erreichbar, so kann der HTTP-Betrieb ohne Einschränkungen fortgesetzt werden. Für ein hochverfügbares System muss also insbesondere der Webserver besonders sicher ausgelegt sein, Rabus Server ist nur zu Administrationszwecken notwendig.

Ein Betrieb mit *Rabus* ist nicht notwendig aber sinnvoll, da er die zentrale Überwachung der Geräte vornimmt (insbesondere Batterie-Ladestand) und Konfigurationsänderungen und Firmwareupdates über WLAN erlaubt. Siehe *Rabus Server Handbuch*.

Dieses Dokument beschreibt die Aspekte des HTTP-Betriebs. Für den Betrieb mit Rabus sei auf andere Dokumente verwiesen (*Rabus Server Handbuch, FBConfig Handbuch, Fehlercodes, Rabus Communication Server Interface*).

Kommunikationsablauf HTTP

Bei einem Knopfdruck oder dem Scan eines Barcodes wird eine TCP-Verbindung zum Webserver aufgebaut und ein HTTP-GET (oder HEAD oder POST) mit der konfigurierten URL gesendet. Bei Erhalt der HTTP-Response wird diese interpretiert und die Antwort dargestellt, wenn ein OK-Status (2XX) enthalten ist. Bei einem anderen Status gibt der Call Button Meldung 16 und den entsprechenden Status aus, z.B. 404 für „Seite nicht gefunden“. Anschließend wird die Verbindung wieder abgebaut.

Unabhängig von der HTTP-Kommunikation baut der Call Button – wenn entsprechend konfiguriert – regelmäßig eine Verbindung zum *Rabus Server* auf, um den eigenen Status zu melden. Das Intervall kann in *FBConfig* oder *Rabus Server* festgelegt werden. Ist *Rabus Server* nicht erreichbar, wird keine Meldung angezeigt sondern der Status ignoriert. HTTP-Verbindungen können weiterhin aufgebaut werden. Als Fehlerhinweis wird während des Verbindungsaufbaus „bitte warten… Keine Adminverbindung“ angezeigt.

Um eine Verbindung zu *Rabus Server* außerhalb des eingestellten Intervalls zu bewirken, muss der Knopf für einige Sekunden betätigt werden, bis die Informationsbildschirme (Seriennummer, Konfiguration, Copyright, usw.) angezeigt werden. Beim Loslassen wird dann eine *Rabus*-Verbindung statt der HTTP-Verbindung aufgebaut.

HTTP-Implementation des Call Buttons

Aufgrund der eingeschränkten Möglichkeiten durch die (insbesondere auf Stromverbrauch) optimierte Hardware, sind nur die wichtigsten Aspekte des HTTP-Protokolls realisiert. Durch eine serverseitige Programmierung mit Servlets, o.ä. lässt sich aber die Funktionalität des Call Buttons voll nutzen.

Der Zustand einer Client-Anforderung innerhalb der Anwendung muss auf der Serverseite gespeichert werden, der Call Button ist nicht in der Lage Cookies zu speichern, dynamische URLs zu verwalten oder eine persistente HTTP-Verbindung aufrecht zu erhalten. Jedoch durch die Übermittlung der Call Button ID (Name) in der URL, kann die Applikation eine eindeutige Zuordnung zu einem auf dem Server gemerkten Zustand herstellen. Es wird HTTP, Version 1.0 verwendet.

URL

Die URL wird auf dem Call Button fest konfiguriert und darf bis zu 500 Zeichen Länge haben (relative Pfadangabe zulässig – z.B. /homepage.php... statt http://www.server.com/homepage.php... - da der Host über die IP-Adresse, nicht über DNS angesprochen wird). Die Einstellung geschieht über die serielle Schnittstelle mit *FBConfig* oder über das WLAN mit *RabusAdmin*. Folgende Angaben können durch den Call Button dynamisch in die URL eingefügt werden:

- $N – der konfigurierte Call Button-Name zur Identifikation
- $T – Call Button-Typ: 1 = voco, 2 = Colligo
- $I – Colligo-ID (bei Colligo 0201, 0202, 0203, 0204,… , ansonsten 0101)
- $E – die Art des Ereignisses
    - 1 = Knopfdruck
    - 2 = Barcode gescannt
    - 3 = Timer
    - 4 = externer Eingang (Auto-Call), nur bei entsprechender Sonderausstattung
- $S – der gescannte Barcode
- $B – der Batteriestand in % (nicht-linear)
- $$ – das Dollarzeichen

Bei der Konfiguration werden die obigen Tags in der URL angegeben, Beispiel: Die konfigurierte URL

/homepage.php?name=$N&event=$E&scan=$S

ersetzt der Call Button durch

/homepage.php?name=BUTTON0001&event=2&scan=1234567890

HTTP Header in der Antwort

Folgende HTTP-Header werden verarbeitet wenn in der Webserver-Antwort vorhanden:

a) Display-Inhalt

Folgende 4 Bezeichner geben die 4 Zeilen des Displays an:

sachnummer:
benennung:
zeitpkt:
meldungstext:

Die Ausgabe erfolgt in US-ASCII plus den deutschen Umlauten aus der Codepage OEM850 (Ä = 8E hex, ä = 84, Ö = 99, ö = 94, Ü = 9A, ü = 81, ß = E1). Es stehen pro Zeile 20 Zeichen zur Verfügung.

b) Lampensteuerung

Der Bezeichner

lampenstatus:

gibt den Fehlerstatus seitens der Applikation an und steuert damit die LEDs. Folgende Stati sind möglich:

- 0 kennzeichnet keinen Fehler („Bestellung akzeptiert“) - grüne LED leuchtet (nicht bei Wakeup) – Defaultzustand, wenn kein Lampenstatus angegeben

- 1 bedeutet Fehler („Bestellung nicht akzeptiert“) - rote LED leuchtet (auch bei Wakeup)

- 255 erzeugt eine HTTP-Fehler (#16) – zu Testzwecken

- Ansonsten gilt folgendes Bitmuster (angegeben als Dezimalzahl):

    - Bit 0-1: Reserviert
        - Immer 11

    - Bit 2-4: Frequenz
        - 000: aus
        - 001: 0,5Hz
        - 010: 1Hz
        - 011: 2Hz
        - 100: 3Hz
        - 101: 5Hz
        - 110: Reserviert
        - 111: Konstant an

    - Bit 5: Reserviert
        - Immer 0

    - Bit 6-7: LED
        - 00: Reserviert
        - 01: rote LED
        - 10: grüne LED
        - 11: rote und grüne LED

Wertetabelle:

LED	**Aus**	**Konstant an**	**Blinkend, 0,5Hz**	**Blinkend, 1Hz**	**Blinkend, 2Hz**	**Blinkend, 3Hz**	**Blinkend, 5Hz**
Grün	195	159	135	139	143	147	151
Rot	195	95	71	75	79	83	87
Grün + Rot	195	223	199	203	207	211	215

Die Dauer der Signalisierung wird – für rote und grüne LED unabhängig – durch die Konfiguration des Call Buttons bestimmt. Für 0 und 1 wird in der Konfiguration auch der Blinkzustand (dauerhaft oder blinkend) angegeben.

Der Bezeichner

rcode:

sofern vorhanden, übersteuert den Lampenstatus. Wenn rcode vorhanden und ungleich 0, wird ein Fehlerstatus signalisiert (wie *lampenstatus: 1*)

c) Timer

Durch den Bezeichner

wakeup:

wird der Timer eingestellt, sodass sich der Call Button nach der angegebenen Zeit mit der konfigurierten URL wieder beim Webserver meldet. Die Angabe von $E in der URL bewirkt die Übergabe des Ereignistyps, der in diesem Falle 2 ist, sodass die Applikation dieses Ereignis von einem Knopfdruck (1) oder Scan (3) unterscheiden kann.

Die Zeitangabe erfolgt mit angefügtem s (5 bis 255 Sekunden) oder m (1 bis 255 Minuten). Beispiele: 10s für 10 Sekunden, 1m für 1 Minute.

Im Falle eines Verbindungsfehlers bei einem Ereignis wird nach 1 Minute ein neues Wakeup durchgeführt, wenn bei der letzten korrekt empfangenen HTTP-Response eine Wakeup-Zeit mitgegeben wurde. Bei wiederholten Fehlern wird die Zeit jeweils verdoppelt, bis 255 Minuten erreicht sind. Ab dann wird jeweils in 255-Minuten-Intervallen erneut kommuniziert. Dies lässt sich nur durch eine HTTP-Response ohne Wakeup-Zeit verhindern, oder indem man den Call Button abschaltet (Taste drücken, bis Copyright erscheint – dann loslassen. Im Display erscheint dann – AUS –).

Colligo

Der Satelliten-Knopf Colligo verwendet über HTTP dasselbe Protokoll wie voco. Zur Identifizierung der Colligos wird die ID über den Platzhalter $I dynamisch in die URL eingefügt. Colligo kann mit der IDs 21...29, 31...39 oder 41...49 konfiguriert werden. Ist bei einem Colligo die ID 21 eingestellt, so wird statt $I „0201“ übergeben, bei ID 22, „0202“, bei ID 49, „0409“, usw. Wird stattdessen der Knopf am voco betätigt, wird „0101“ übergeben.

Die HTTP-Response wird bei einem Colligo Call ähnlich einem voco Call verarbeitet – jedoch mit einigen Einschränkungen. Die rote und grüne LED des Colligo werden entsprechend (Lampenstatus = 0 oder = 1) angesteuert, allerdings ist die Blinkfrequenz nicht beeinflussbar.

Das 4-stellige LC-Display ist nicht frei steuerbar. Im Falle eines guten Abrufs (Lampenstatus = 0), wird in der Folge Tag und Uhrzeit des letzten Abrufs im Wechsel angezeigt. Bei Lampenstatus = 1 wird „nEIn“ angezeigt, bei einem Netzwerkfehler „FEHL“. Zuvor wird „CALL“ angezeigt, wenn der Knopfdruck festgestellt wurde und „SEnd", sobald der Abruf kommuniziert wird.

Installation von Rabus (optional)

Bei einer Erstauslieferung muss zunächst die Software installiert werden:

- Webserver und Web-Applikation
- Rabus Server auf Server-PC (siehe *Rabus Server Handbuch*)
- Administrationstools (*RabusAdmin*, *RabusMonitor* (siehe *Rabus Server Handbuch*), sowie *FBConfig* (siehe *FBConfig Handbuch*) auf dem Administrations-PC

Nach der Installation von Rabus Server wird dieser mit RabusAdmin eingestellt. Dabei muss darauf geachtet werden, dass in dem Formular „Application Server Settings“ die mittlere Option gewählt wird („Direct HTTP connection“).

Neue Call Buttons werden über ein serielles RS232-Kabel mit dem Administrations-PC verbunden und durch FBConfig eingstellt. Für die HTTP-Kommunikation müssen neben den üblichen Netzwerk- und WEP-Einstellungen folgende Parameter eingestellt werden:

- Registerkarte HTTP: HTTP-Protokoll aktiveren, URL angeben

- Registerkarte Netzwerk: Neben der Administrationsserver-Verbindung muss auch der HTTP-Server angegeben werden: Host-IP zum Webserver, Port (normalerweise 80), ggs. Router zum Webserver