

Rabus CallButton Server

Handbuch

zum *Rabus Administrations-Server* Version 2.12

© Copyright by Detlef Reil, 2003-2010

Inhalt

1	Funktionalität	2
2	Anwendungskommunikation	2
2.1	CallButton-Kommunikation mit Rabus Server	2
2.2	HTTP-Kommunikation mit Webserver	3
3	Rabus Verwaltungsaufgaben.....	3
3.1	Alert Server.....	3
3.2	Ereignis-Logging	4
3.3	Verteilung von Firmware-Updates	4
3.4	Globale CallButton-Konfiguration (Automatisches Konfigurations-Update)	4
3.5	CallButton-Fehlerhistorie	5
3.6	CallButton-Gruppen	6
4	Installation	6
4.1	Rabus-Dienst	6
4.2	Die Rabus INI-Datei.....	8
4.3	E-Mail-Installation.....	11
4.4	Tools	12
5	RabusAdmin.....	13
5.1	Mit Rabus Server verbinden	13
5.2	Navigation	13
5.3	Login.....	13
5.4	Server-Informationen.....	13
5.5	CallButton-Gruppenzuordnung	14
5.6	CallButton Ansicht	15
5.7	Ansicht Event History	16
5.8	Ansicht Call Button Config	17
5.8.1	Ansicht Network	19
5.8.2	Ansicht WEP	20
5.8.3	Ansicht HTTP	20
5.8.4	Ansicht Miscellaneous	21
5.9	Ansicht Firmware Update Settings.....	22
5.10	Komponente Alert Server	22
5.10.1	Ansicht Alert Event Log	22
5.10.2	Ansicht Alert Server Settings	24
5.11	Applikations-Demomodus	25
6	RabusMonitor.....	28
6.1	Mit Rabus Server verbinden	28

1 Funktionalität

Der *Rabus CallButton Server* ist die zentrale Instanz zur Verwaltung der CallButtons (Überwachung, Konfiguration und Firmware Updates). Er überwacht die Funktion der CallButtons und kann einen Alarm auslösen, wenn eine Fehlfunktion vorliegt.

Der *Rabus Server* enthält zudem eine Kommunikations-Server-Komponente für proprietäre Anwendungen. Er puffert die Kommunikation zwischen Anwendung und CallButtons und verteilt die Information zwischen beiden.

2 Anwendungskommunikation

Die Kommunikation zwischen CallButtons und der Applikation kann auf zwei verschiedene Arten konfiguriert werden:

- 1.) über *Rabus Server* und das proprietäre Rabus-Protokoll. Die Anwendung verbindet sich über TCP/IP mit *Rabus Server* und wird durch Rabus über CallButton-Ereignisse benachrichtigt. Sie sendet eine Antwort zurück, und Rabus puffert die Information und verteilt sie an den entsprechenden CallButton.
- 2.) Über einen Standard-Webserver und das HTTP-Protokoll. Die Anwendung ist im Webserver integriert und durch Java Servlets, Perl / PHP, Coldfusion, ASP, o.ä. realisiert.

2.1 CallButton-Kommunikation mit *Rabus Server*

Der CallButton ist aus Stromspargründen im Ruhezustand abgeschaltet (außer LC-Display und Überwachungsschaltung). Beim Auftreten eines der folgenden Ereignisse wacht er auf und nimmt Verbindung zu *Rabus Server* durch TCP/IP über WLAN auf:

- Knopf betätigt ("T")
- Barcode gescannt ("S")
- Externer Eingang ("A")
- Wecker ("W"; alle 5 Sekunden bis 255 Minuten)
- Firmware-Update über TCP/IP erledigt ("U")
- Konfigurationsänderung über TCP/IP erledigt ("C")

Nach dem Senden eines Kommunikationstelegramms für das entsprechende Ereignis wartet der CallButton auf eine Antwort von *Rabus* mit dem neuen Anzeige-Inhalt und Lampenzustand. Nach Erhalt schaltet er wieder ab. Während der CallButton abgeschaltet ist, kann er nicht mit dem Host kommunizieren. Eine Kommunikation wird also immer durch den CallButton initiiert und nie durch den Server.

Rabus puffert jedes CallButton-Ereignis und ist für die Applikation jederzeit ansprechbar. Er sendet neue Anzeige-Inhalte der Anwendung zum CallButton, sobald der die Verbindung aufnimmt.

Die Applikation kommuniziert mit *Rabus Server* über TCP/IP mit einem speziellen proprietären Protokoll mit einer Anzahl von Befehlen:

- GET – Informationen zu einem bestimmten CallButton anfordern
- ALL – Informationen zu allen bekannten CallButtons anfordern
- ENE – die automatische Benachrichtigung über CallButton-Ereignisse einschalten
- DIS – ENE abschalten
- SET – Anzeige-Informationen und Lampenzustand zu einem CallButton vorgeben

Details sind in der *Rabus Interface Documentation* beschrieben.

2.2 HTTP-Kommunikation mit Webserver

Die CallButtons können alternativ bei folgenden Ereignissen eine direkte HTTP-Verbindung zu einer Applikation in einem Standard-Webserver aufnehmen:

- Knopf betätigt
- Barcode gescannt
- Externer Eingang
- HTTP-Timer

Rabus Server ist in diesem Fall nicht mehr für die Kommunikation zuständig, wird aber trotzdem zu Administrationszwecken benötigt und bearbeitet alle anderen als die obigen Ereignisse. Siehe Beschreibung „*HTTP Systembeschreibung*“.

3 Rabus Verwaltungsaufgaben

3.1 Alert Server

Rabus Server überwacht die korrekte Funktion der CallButtons und kann Abweichungen aufnehmen und als Alarm an die Administratoren melden:

- Seit einiger Zeit keine Kommunikation mehr mit einem CallButton (z.B. weil dieser defekt ist). Da alle CallButtons regelmäßig Verbindung mit *Rabus Server* aufnehmen, gilt es als fehlerhafter Zustand, wenn sich ein Gerät für längere Zeit nicht meldet.
- CallButton Batterieladestand unterschreitet einen bestimmten Wert. Um die Batterie rechtzeitig wechseln zu können, kann über diese Funktion ein Alarm ausgelöst werden
- Keine gültige CallButton Client-Lizenz. Die Lizenz von *Rabus Server* ist auf eine bestimmte Anzahl Geräte begrenzt.

- Hardware-Key wurde entfernt. Der Hardware-Key enthält die Lizenz-Informationen ist zum Betrieb von *Rabus Server* notwendig.
- Server Start / Ende. Wenn der Rabus Server gestartet oder beendet wird, wird durch diese Funktion ein Alarm ausgelöst.
- Ein neuer, bisher unbekannter CallButton hat sich gemeldet

Alle Alarm-Ereignisse werden in einer Liste gespeichert und können an eine Logdatei angefügt werden oder eine E-Mail an einen Administrator senden.

3.2 Ereignis-Logging

Alle CallButton-Ereignisse zu *Rabus Server* werden zu Zwecken der späteren Analyse aufgenommen. Die Anzahl der gespeicherten Ereignisse kann in der INI-Datei eingestellt werden.

3.3 Verteilung von Firmware-Updates

Rabus Server kann die Verteilung und Aktivierung von neuen Firmware-Versionen für die CallButtons vornehmen. Er überwacht bei jeder Kommunikation die aktuelle Version der Einheiten und initiiert einen Firmware-Update, wenn eine neuere Version verfügbar ist.

Normalerweise werden dabei alle CallButtons einer Gruppe berücksichtigt, die sich mit *Rabus Server* verbinden. Über zwei Steuerdateien kann die Menge jedoch eingeschränkt werden. Befindet sich die Datei

`AutoUpdateInclude.txt`

im Programmverzeichnis (z.B.: `c:\Programme\Rabus Communication Server`), so werden nur jene CallButtons berücksichtigt, deren Namen in der Aufzählung in dieser Datei enthalten sind, alle anderen nicht. Ist die Datei nicht vorhanden, so werden alle CallButtons berücksichtigt. Befindet sich dort die Datei

`AutoUpdateExclude.txt`

so werden die CallButtons, deren Namen in der Liste stehen, nicht berücksichtigt.

3.4 Globale CallButton-Konfiguration (Automatisches Konfigurations-Update)

Globale Konfigurationsänderungen können an die CallButtons verteilt werden (Konfigurationsupdate, AutoConfig). *Rabus Server* überwacht die Konfigurationsversion bei jeder Kommunikation und stellt ausgewählte Parameter ein, wenn eine neue Version verfügbar ist.

Jedoch muss diese Funktion mit großem Fingerspitzengefühl vorgenommen werden, damit *Rabus* über das Netzwerk für alle Einheiten erreichbar bleibt. Jeder CallButton nimmt mindestens alle 4,5 Stunden Verbindung zu *Rabus* auf, so dass es eben bis zu 4,5 Stunden dauern kann, bis eine Änderung bei allen Einheiten durchgeführt wurde.

Wenn Netzwerk-Parameter verändert werden sollen (z.B. Server-IP-Adresse, WLAN-ESS-Id, WEP-Schlüssel, etc.) muss eine doppelte Infrastruktur vorhanden sein, falls das System durchgehend verfügbar bleiben soll, weil während der Verteilung einige Einheiten noch die alten Werte besitzen, während andere bereits neue Werte aktiviert haben. Nachdem alle CallButtons auf die neue Konfigurationsversion aktualisiert wurden, kann die alte Infrastruktur abgeschaltet werden. Durch *Rabus Monitor* kann jederzeit eingesehen werden, welche Einheiten noch nicht aktualisiert wurden.

Wenn eine doppelte Infrastruktur nicht zur Verfügung steht und die durchgehende Verfügbarkeit nicht notwendig ist, muss die alte Infrastruktur beibehalten werden, bis alle Einheiten aktualisiert wurden.

Wenn die CallButtons durch fehlerhafte Einstellungen die Netzwerkverbindung zu *Rabus Server* verlieren, lassen sie sich nur noch jeweils mittels *FBConfig* über ein Serielles Kabel konfigurieren. Das kann bei einer großen Installation sehr ärgerlich sein!

Die Einrichtung des Konfigurationsupdates wird im Kapitel **Ansicht Call Button Config** weiter unten beschreiben.

Normalerweise werden beim automatischen Konfigurationsupdate alle CallButtons einer Gruppe berücksichtigt, die sich mit *Rabus Server* verbinden. Über zwei Steuerdateien kann die Menge jedoch eingeschränkt werden. Befindet sich die Datei

```
AutoConfigInclude.txt
```

im Programmverzeichnis (z.B.: c:\Programme\Rabus Communication Server), so werden nur jene CallButtons berücksichtigt, deren Namen in der Aufzählung in dieser Datei enthalten sind, alle anderen nicht. Ist die Datei nicht vorhanden, so werden alle CallButtons berücksichtigt. Befindet sich dort die Datei

```
AutoConfigExclude.txt
```

so werden die CallButtons, deren Namen in der Liste stehen, nicht berücksichtigt.

3.5 CallButton-Fehlerhistorie

Ab der CallButton-Version 071119 wird auf den Geräten eine Fehlerhistorie mitgeführt. Diese kann mit *Rabus Server* ab Version 1.80 angefordert und in der CallButton-Ansicht in *RabusAdmin* (ab Version 1.80) dargestellt werden. Dazu muss in der INI-Datei der Parameter `GetLog = 1` gesetzt werden (entspricht dem Defaultwert, sodass der ganze Eintrag weggelassen werden kann).

Jeder CallButton speichert die letzten 8 HTTP-Kommunikationsfehlerereignisse. Außerdem wird der letzte Rabus-Kommunikationsfehler gespeichert. Die Fehlereinträge werden durch eine Rabus-Kommunikation abgeholt und gelöscht.

Ist der Parameter `WriteLog = 1` gesetzt (Defaultwert), so werden im Unterverzeichnis `.\Error History\` pro CallButton Log-Dateien mit der Fehlerhistorie angelegt (z.B.: c:\Programme\Rabus Communication Server\Error History\voco001.log). Inhalt: Zeitstempel; Fehlernummer; (HTTP) oder (Rabus).

3.6 CallButton-Gruppen

Die CallButtons können unterschiedliche Gruppen mit unterschiedlichen Konfigurationssätzen zugeordnet werden. Neue CallButtons werden in die Gruppe DEFAULT gestellt und können dann mit *RabusAdmin* anderen Gruppen zugeordnet werden.

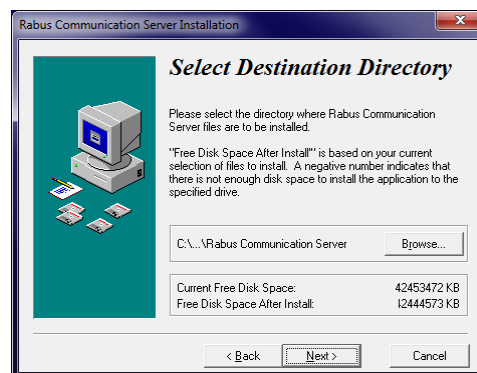
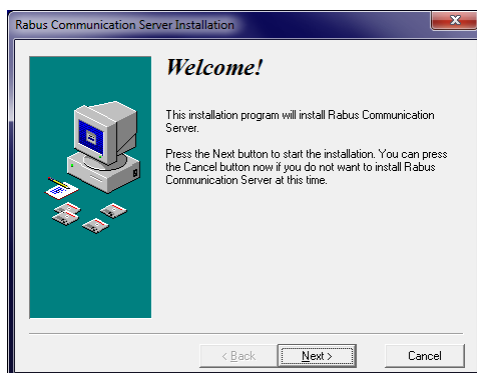
4 Installation

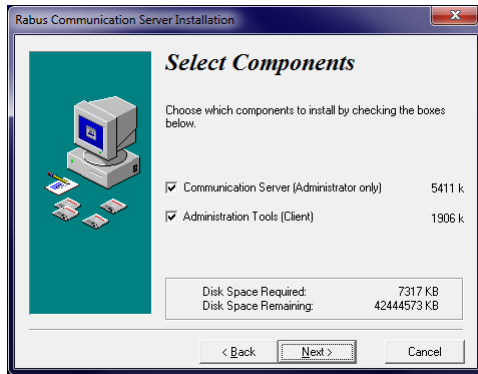
4.1 Rabus-Dienst

Rabus Server läuft als Dienst auf einem PC mit:

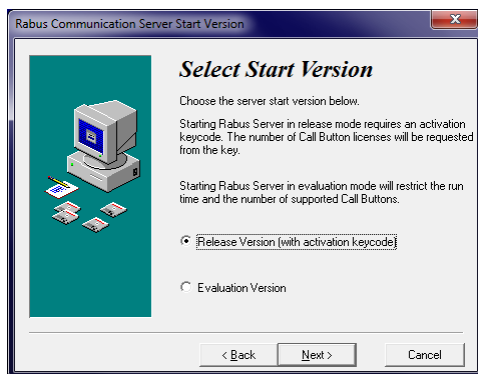
- Windows 2000 Professional oder Server
- Windows XP Professional
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7

Die Installation kann nur mit Administrator-Rechten vorgenommen werden und kopiert alle notwendigen Dateien und installiert den Dienst. Um die Installation zu starten, rufen Sie einfach die *SETUP.EXE* auf der CD auf und beantworten die Fragen in den Dialogen.

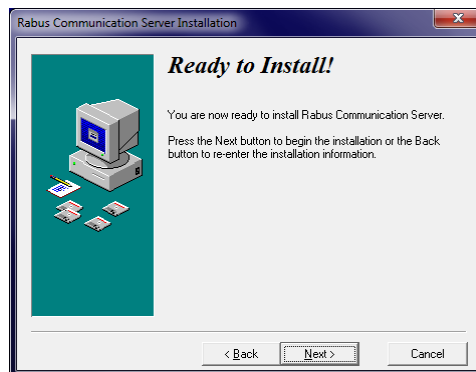
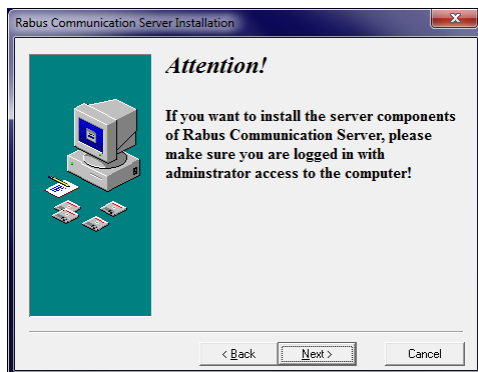




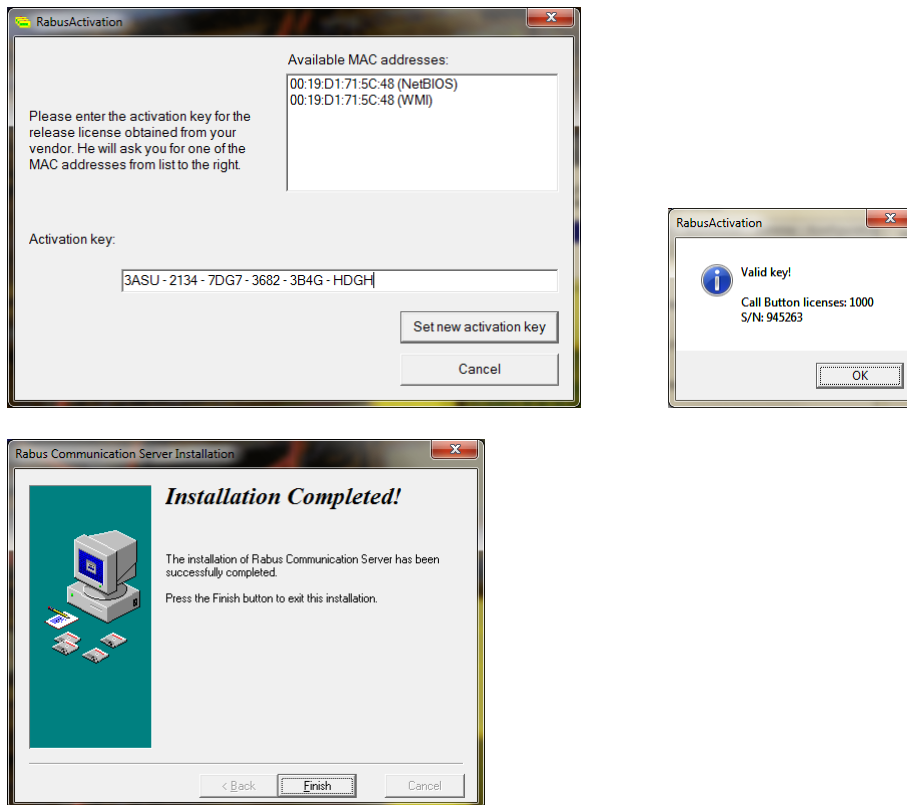
Durch Auswahl von “Communication Server” wird *Rabus Server* installiert (Sie müssen mit Administrator-Rechten eingeloggt sein!). Durch Auswahl von “Administration Tools” werden die Programme *Rabus Administration Tool* und *Rabus Monitor Tool* installiert. Auf Client-PCs mit Netzwerk-Zugriff auf den Serverrechner können auch nur die Verwaltungsprogramme eingerichtet werden.



Die Evaluationsversion ist nur für Testzwecke gedacht und erfordert keinen Lizenzschlüssel oder Hardware-Dongle. Sie ist beschränkt auf maximal 5 CallButtons, 5 Stunden kontinuierlicher Betriebszeit und einem speziellen Ablaufdatum.



Wenn die Release-Version mit Aktivierungsschlüssel installiert wurde, wird der Lizenzschlüssel abgefragt.



Wenn notwendig, kann anschließend die Anpassung der internen Rabus-Einstellungen mit einem Texteditor an der Datei `RabusCommSrv.ini` vorgenommen werden. Diese Änderungen werden erst nach dem Neustart des Dienstes (oder des PCs) wirksam.

Die Verwaltung und Konfiguration der normalen Betriebsparameter von *Rabus* wird dann mit den *Rabus Administration Tools* vorgenommen.

4.2 Die *Rabus* INI-Datei

In der *Rabus* INI-Datei werden die grundsätzlichen Einstellungen für den *Rabus Server* vorgenommen. Änderungen an dieser Konfigurationsdatei werden erst übernommen, wenn der *Rabus Server* Dienst neu gestartet wird. Die Konfigurationsdatei kann mit einem standardmäßigen Texteditor bearbeitet werden (z.B. Notepad) und ist in verschiedene Sektionen aufgeteilt, die jeweils durch eine Zeile mit einem Bezeichner in eckigen Klammern eingeleitet werden. Alle Zeilen, die mit einem Semikolon eingeleitet werden, werden als Kommentarzeilen ignoriert.

Folgende Parameter werden benutzt. Für fehlende Angaben werden ihre Standardwerte angenommen.

- Sektion [CallButtons]
 - TCPPort = 8000
 - TCP-Port für die CallButton-Verbindungen. Dieser muss mit der Einstellung der CallButtons übereinstimmen
 - DefaultStayAwakeTime = 0
 - Wartezeit (in Sekunden) für die der CallButton eingeschaltet nach einem Ereignis bleibt.

Nach einem Ereignis meldet Rabus zunächst an die Applikation und sendet dann normalerweise die gespeicherten Anzeigeinformationen zum CallButton. Dieser schaltet daraufhin sofort ab. Durch die DefaultStayAwakeTime bleibt der CB eine Zeit eingeschaltet, so dass die Applikation Zeit hat, eine neue Antwort zu generieren und zu senden.

Wenn Rabus durch das SET Kommando (3. Parameter) instruiert wurde, bei einem Ereignis auf die Applikation zu warten, sendet er nichts zum CB bis die Applikation sich zurückmeldet. Dauert dies zu lange, greift die im CallButton konfigurierte Anbleibzeit und löst eine Fehlermeldung im CB aus.

- DisableCBConfig = 0
 - wenn auf 1 gesetzt, werden die Konfigurationsparameter der CallButtons nicht abgefragt. Dadurch kann die automatische CallButton-Konfiguration nicht stattfinden.
- NoApplicationMessage = “====RABUS SOFTWARE==== Application is not running!”
 - diese Meldung wird auf dem CallButton nach einem Ereignis angezeigt, wenn keine Applikation mit Rabus verbunden ist. Der CallButton hat eine Anzeige von 20x8 Zeichen – die Zeilen müssen zur Ausrichtung mit Leerzeichen aufgefüllt werden.
- NoApplicationWakeupTime = 255
 - Weckzeit (in Minuten), nach der CallButton nach der obigen Meldung eine neue Verbindung versucht
- CBNotInUseMessage = “====RABUS SOFTWARE==== Call Button is not in use!”
 - diese Meldung wird auf dem CallButton nach einem Ereignis angezeigt, wenn keine Anzeigeinformation für dieses Gerät vorliegt, weil sie durch die Applikation noch nicht vorgegeben wurde.
- CBNotInUseWakeupTime = 255
 - Weckzeit (in Minuten), nach der CallButton nach der obigen Meldung eine neue Verbindung versucht
- LicenseTimeExceedMessage = “====RABUS SOFTWARE==== License time expired!”
 - diese Meldung wird auf dem CallButton nach einem Ereignis angezeigt, wenn die Laufzeit der Evaluationsversion von Rabus abgelaufen ist.
- LicenseCBExceedMessage = “====RABUS SOFTWARE==== Number of Call Button licenses exceeded!”
 - diese Meldung wird auf dem CallButton nach einem Ereignis angezeigt, wenn die maximale Anzahl der Client-Lizenzen erreicht ist.
- HardwareKeyMessage = “====RABUS SOFTWARE==== Hardware key not present!”
 - diese Meldung wird auf dem CallButton nach einem Ereignis angezeigt, wenn der Hardware-Key entfernt wurde.
- NoHardwareKeyWakeupTime = 255
 - Weckzeit (in Minuten), nach der CallButton nach der obigen Meldung eine neue Verbindung versucht
- UpdateMessage = “====RABUS SOFTWARE==== !Waiting for update!”
 - diese Meldung wird auf dem CallButton während eines Firmware-Updates durch Rabus angezeigt.

- ConfigMessage = “====RABUS SOFTWARE==== !Waiting for config!”
- diese Meldung wird auf dem CallButton während eines Konfigurations-Updates durch Rabus angezeigt.
- NoDemoFileMessage = “====RABUS SOFTWARE==== Demo file not found!”
- diese Meldung wird auf dem CallButton nach einem Ereignis angezeigt, wenn der Applikations-Demomodus aktiviert ist und keine passende Demodatei vorhanden ist.
- DemoFileErrorMessage = “====RABUS SOFTWARE==== Demo file access error!”
- diese Meldung wird auf dem CallButton nach einem Ereignis angezeigt, wenn der Applikations-Demomodus aktiviert ist und die passende Demodatei nicht geöffnet werden kann.
- GetLog = 1
- von CallButtons mit einer Version ab 071119 wird bei jeder Rabus-Kommunikation das Fehlerprotokoll und das aktuelle Display angefordert
- WriteLog = 1
- das über GetLog abgeholte Fehlerprotokoll wird im Unterverzeichnis „\Error History“ unter dem Namen des CallButtons in einer Logdatei gespeichert
- Sektion [Application]
 - TCPPort = 18305
- TCP-Port für die Verbindung von der Applikation
 - ProtocolTimeout = 10
- Jedes Telegramm von der Applikation zu *Rabus Server* muss vollständig in dieser Zeit (in Sekunden) gesendet werden
 - ResponseTimeout = 20
- Jede Antwort von der Applikation muss innerhalb dieser Zeit (in Sekunden) gesendet werden
- Sektion [Admin]
 - TCPPort = 18301
- TCP-Port für die Verbindung von den Administrationstools *RabusAdmin* and *RabusMonitor*
 - Password = “0000” - default: keins
- dieses Passwort muss für Konfigurationsänderungen in *RabusAdmin* eingegeben werden
- Sektion [System]
 - EvaluationMode = 0
- wenn auf 1 gesetzt, wird *Rabus Server* im eingeschränkten Evaluationsmodus betrieben. Dazu ist kein Hardware-Key notwendig
 - CheckDongle = 1
- wenn auf 1 gesetzt (oder nicht vorhanden), wird der Hardware-Key überprüft

- WorkingDir = “c:\temp” - default: Installationsverzeichnis (z.B.: “c:\Programme\Rabus Communication Server”)
 - Verzeichnis für Log-, Konfigurations- und Datenbankdatei, etc.
- Sektion [Log]
 - Level = % 00010101 01010010 - default: 0
 - (Binärangabe) Detaillevel der Einträge in der Logdatei
 - Bit0 (letztes Bit): nicht benutzt
 - Bit1: Verbindungsauf- und Abbau der Applikation anzeigen
 - Bit2: Verbindungsauf- und Abbau von RabusAdmin/RabusMonitor anzeigen
 - Bit3: Verbindungsauf- und Abbau von der CallButtons anzeigen
 - Bit4: CallButton-Kommunikationstelegramme anzeigen
 - Bit5: Details zur CallButton-Kommunikation anzeigen
 - Bit6: Lizenz-Fehler anzeigen
 - Bit7: Applikationskommunikation anzeigen
 - Bit8: Applikationskommunikationsfehler anzeigen
 - Bit9: Applikationskommunikationsdetails anzeigen
 - Bit10: CallButton Auto-Update/-Config anzeigen
 - Bit11: Kommunikationsdetails von RabusAdmin / RabusMonitor anzeigen
 - Bit12: Kommunikationsfehler von RabusAdmin / RabusMonitor anzeigen
 - Bit13: Kommunikation von RabusAdmin / RabusMonitor anzeigen (Bit7 muss auch gesetzt sein)
 - Bit14 (erstes Bit): CallButton-Protokoll anzeigen
 - Bei einem Wert von 0 wird keine Log-Datei erzeugt
 - HistoryDays = 30 - default: kein Löschen
 - Logdateien werden gelöscht, wenn sie älter als die Zeit (in Tagen) sind
- Sektion [Demo]
 - AppDemoDir = “.\FBDemo”
 - Unterverzeichnis für die Textdateien des Applikations-Demomodus. Gilt nur, wenn die Option “Retrieve from files” gesetzt ist (siehe *RabusAdmin*).

4.3 E-Mail-Installation

Wenn die E-Mail-Benachrichtigung von Alarmen gewünscht ist, muss ein E-Mail-Programm mit MAPI-Schnittstelle auf dem Server installiert sein, z.B. Outlook Express. Outlook aus dem Microsoft Office lässt leider kein automatisches Senden von E-Mails mehr zu, so dass es ungeeignet ist. In der Einrichtung des E-Mail-Programms muss das Versenden von Mails durch Programme erlaubt werden und das sofortige Senden eingeschaltet werden.

Ein gültiges Benutzerkonto muss im E-Mail-Programm eingerichtet werden, und der Rabus-Server-Dienst muss dieses Konto verwenden. Dazu muss der Dienst nach der Installation umgemeldet werden. In der Management-Konsole (Systemsteuerung / Verwaltung / Dienste) muss in den Eigenschaften des Dienstes „RabusCommSrv“, Registerkarte „Anmelden“ nicht das „Lokale Systemkonto“ sondern das entsprechende Benutzerkonto ausgewählt werden. Nach einem Neustart des Dienstes sollte das Senden von E-Mails möglich sein.

4.4 Tools

RabusAdmin und *RabusMonitor* können auf dem Server-PC und/oder auf Client-PCs mit Netzwerkzugriff installiert werden.

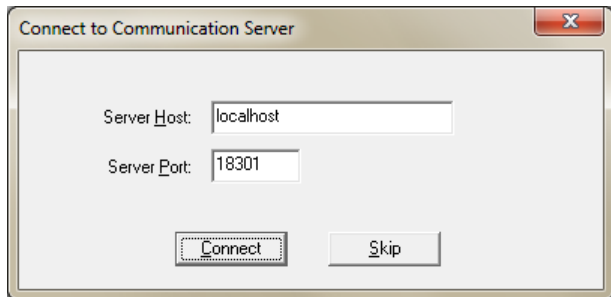
Folgende Betriebssysteme werden unterstützt:

- Windows 2000 Professional oder Server
- Windows XP Home oder Professional
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7

5 RabusAdmin

5.1 Mit *Rabus Server* verbinden

Beim Start fragt *RabusAdmin* zunächst die Verbindungsangaben zu *Rabus Server* ab (Hostname oder IP-Adresse und TCP-Port, standardmäßig 18301) und baut anschließend die Verbindung auf. Die Angaben werden für zukünftige Aufrufe gespeichert.



5.2 Navigation

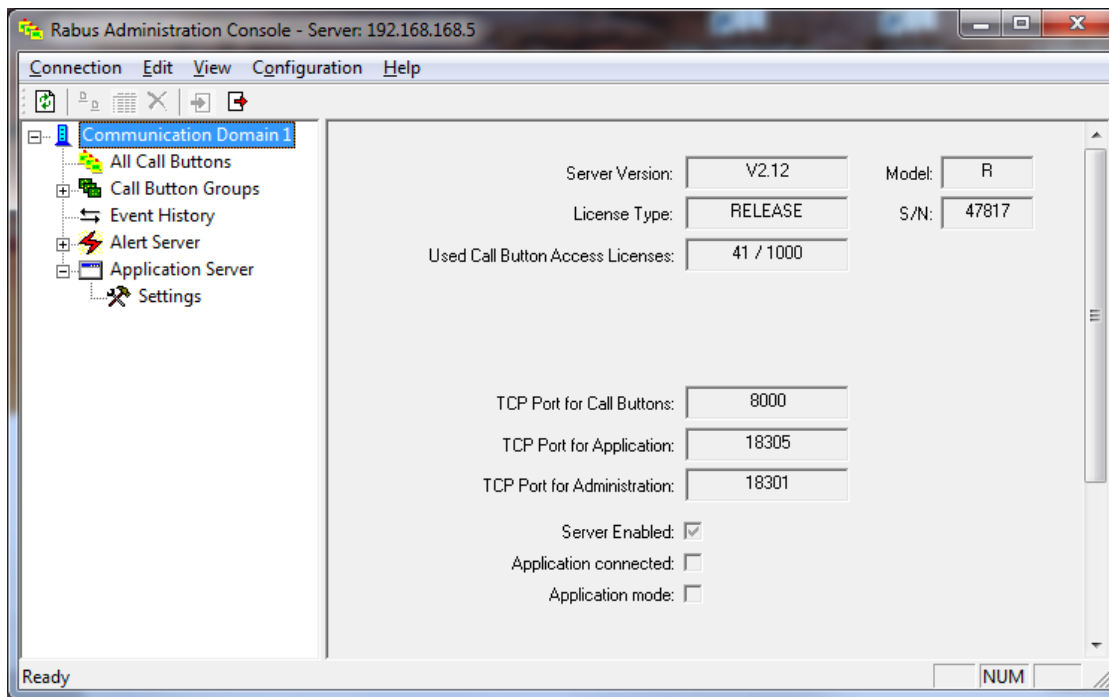
Die Baumansicht auf der linken Seite steuert die verschiedenen Serverkomponenten an.

5.3 Login

Um Einstellungen ändern zu können, müssen Sie sich einloggen. Wählen Sie den Menüpunkt "Configuration" / "Login" oder Klicken Sie auf das Türsymbol. Das anzugebende Passwort ist in der *Rabus Server* INI-Datei definiert.

5.4 Server-Informationen

Durch Auswählen der Komponente **Communication Domain 1** wird auf der rechten Seite eine Übersicht über den *Rabus Server* angezeigt.



- *Rabus Server* Version und Modell
- Lizenzart (RELEASE oder EVALUATION), Seriennummer des Hardware-Keys
- Anzahl der benutzten und verfügbaren CallButton-Clientlizenzen
- TCP-Portnummern für die CallButtons, die Applikation und *RabusAdmin* / *RabusMonitor*
- Ob die Server-Kommunikation gestartet ist
- Ob eine Applikation angeschlossen
- Ob der Applikations-Demomodus oder die direkte HTTP-Kommunikation aktiviert ist

Um die Ansicht zu aktualisieren, wählen Sie den Menüpunkt "Server" / "Refresh" oder klicken das linke Symbol in der Werkzeuggeste.

5.5 CallButton-Gruppenzuordnung

Der Baumeintrag **Call Button Groups** enthält die konfigurierten Gruppen (**DEFAULT** und alle zusätzlich konfigurierten Gruppen). Jede benannte Gruppe enthält die CallButton-Liste, die Autokonfigurations-Einstellungen und die Firmwareupdate-Einstellungen. Neue Gruppen werden durch Klick mit der rechten Maustaste auf den Eintrag **Call Button Groups** und Auswahl von "**Add new group**" angelegt. Gruppen können durch Rechts-Klick auf die entsprechende Gruppe und "**Delete group**" gelöscht werden. Die Konfigurationsparameter einer Gruppe können durch Auswahl der Quellgruppe unter "**Copy data from**" kopiert werden.

Ein neuer CallButton wird zunächst automatisch der **DEFAULT**-Gruppe zugeordnet und kann durch Rechts-Klick auf den CallButton in der rechten Liste und Auswahl der Zielgruppe unter "**Assign to group**" verschoben werden.

5.6 CallButton Ansicht

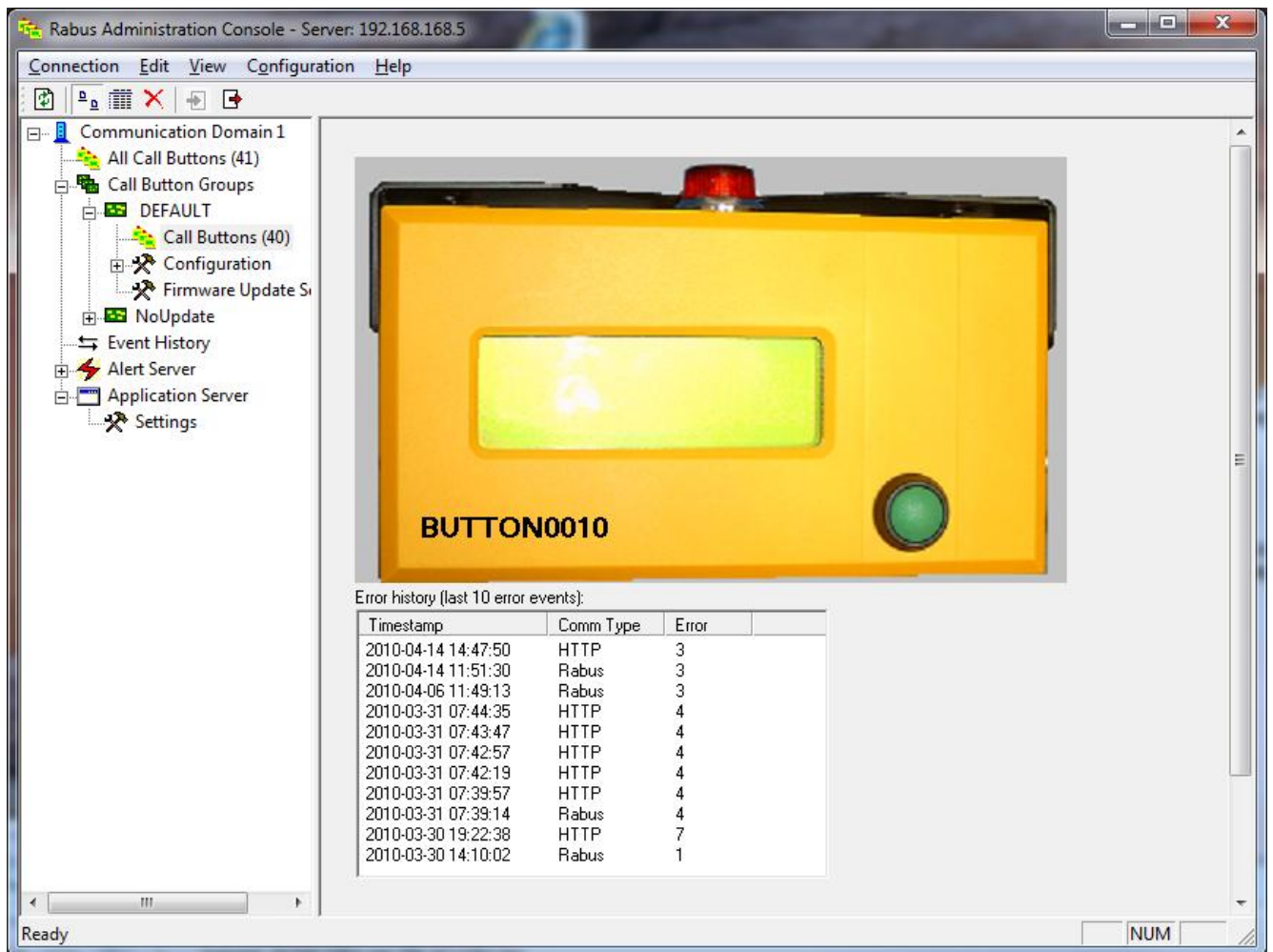
Durch Auswahl der Komponente **Call Buttons** erscheint auf der rechten Seite eine Liste der bekannten CallButtons mit dem jeweils aktuellen Status. Diese Ansicht ist ähnlich der Liste in *RabusMonitor* aufgebaut. Durch Klicken auf die Spaltenüberschrift wird die Liste nach der entsprechenden Spalte sortiert. Durch Ziehen des Spaltenseparators kann die Spalte vergrößert, verkleinert oder versteckt werden.

Im Gegensatz zu *RabusMonitor* werden die Zeilen nicht automatisch aktualisiert. Dazu wählen Sie den Menüpunkt "Server" / "Refresh" oder klicken das linke Symbol in der Werkzeugleiste.

Group	Type	Name	Cmd	B...	Last Connect	Wake...	Next Co...	Lamp	Batt	Display
DEFAULT	NT	0000 *	W		2010-04-15 08:08:59				9	
DEFAULT	NT	01141030 *	W		2010-04-21 11:46:21				9	
DEFAULT	NT	01151003 *	T		2010-04-15 14:37:01	5sec	2010-04...	off	9	Call me
DEFAULT	NT	01171001 *	W		2010-04-19 14:11:33				8	
DEFAULT	NT	02141030 *	W		2010-04-21 11:46:45				9	
DEFAULT	NT	02151003 *	W		2010-04-15 14:38:00	4h 15...	2010-04...	off	9	Call me
DEFAULT	NT	03141030 *	W		2010-04-21 11:47:01				9	
DEFAULT	NT	03151003 *	T		2010-04-15 14:35:47	5sec	2010-04...	off	9	Call me
DEFAULT	NT	04141030 *	W		2010-04-21 13:04:36				9	
DEFAULT	NT	04510999 *	W		2010-04-20 16:44:50				2	
DEFAULT	NT	05141030 *	W		2010-04-21 11:47:44				9	
DEFAULT	SE	05310823 *	W		2010-04-19 12:03:42				9	
DEFAULT	NT	06141030 *	W		2010-04-21 11:47:58				9	
DEFAULT	NT	07141030 *	W		2010-04-21 11:48:03				9	
DEFAULT	NT	08141030 *	W		2010-04-21 11:48:30				9	

Zu den Details der Angaben in der Liste lesen Sie bitte das Kapitel zu *RabusMonitor*.

Durch Doppelklicken einer Zeile in der linken oder rechten Ansicht oder durch Klicken des dritten Symbols in der Werkzeugleiste erscheint auf der rechten Seite eine Ansicht des entsprechenden CallButtons mit dem aktuellen Anzeige- und Lampenzustand zum Zeitpunkt der letzten Rabus-Kommunikation.



Wird für die Applikationskommunikation HTTP verwendet, ist der dargestellte Anzeigezustand nur bei CallButtons mit einer Version ab 071119 korrekt. Außerdem muss Konfigurationsparameter GetLog = 1 gesetzt sein.

Bei CallButtons mit einer Version ab 071119 wird die Fehlerhistorie angezeigt, wenn Konfigurationsparameter GetLog = 1 aktiviert ist.

5.7 Ansicht Event History

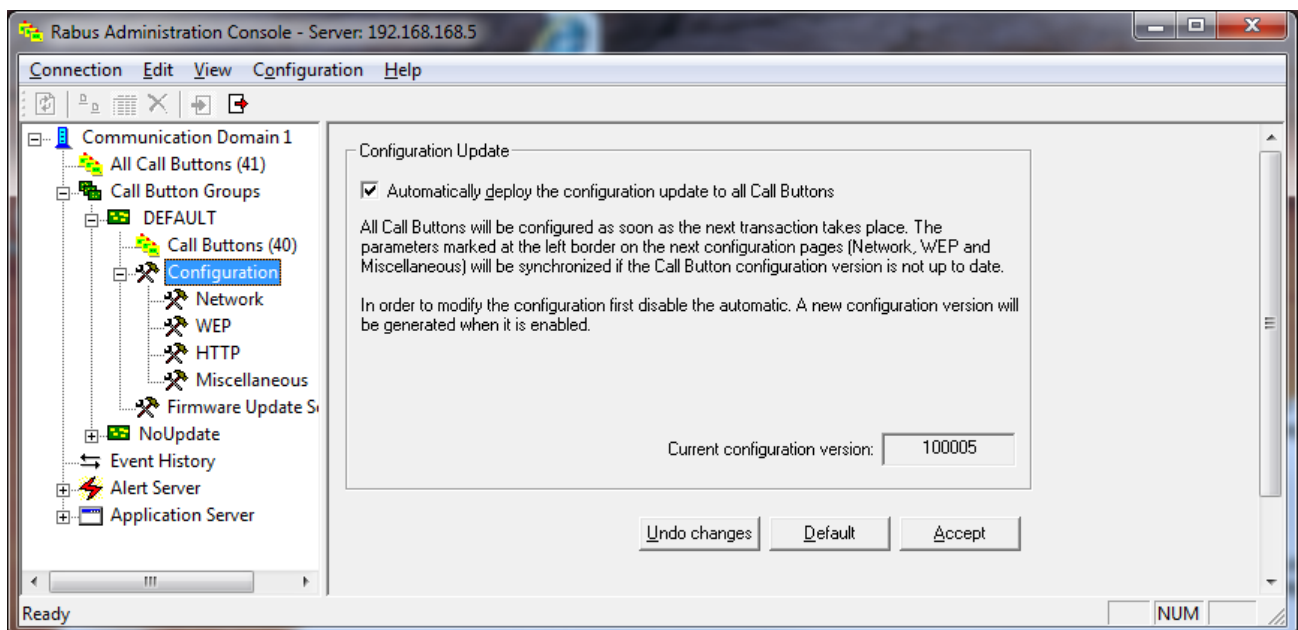
Durch Auswählen der Komponente **Event History** erscheint eine Liste der 200 letzten CallButton-Ereignisse auf der rechten Seite.

Rabus Administration Console - Server: 192.168.168.5

No	Timestamp	Call Button Name	Event...	Barcode
1000	2010-04-21 17:27:52	vocco101701	W	
999	2010-04-21 13:04:36	04141030	W	
998	2010-04-21 12:03:48	vocco083105	W	
997	2010-04-21 11:54:40	22141030	W	
996	2010-04-21 11:50:17	18141030	W	
995	2010-04-21 11:49:56	17141030	W	
994	2010-04-21 11:49:48	16141030	W	
993	2010-04-21 11:49:22	13141030	W	
992	2010-04-21 11:49:06	14141030	W	
991	2010-04-21 11:49:03	12141030	W	
990	2010-04-21 11:48:49	11141030	W	
989	2010-04-21 11:48:36	10141030	W	
988	2010-04-21 11:48:30	08141030	W	

5.8 Ansicht Call Button Config

Über die Ansicht **Call Button Config** mit ihren Unterpunkten wird die automatische CallButton-Konfiguration (AutoConfig, Konfigurationsupdate) eingestellt.



Um in den Unterpunkten (Network, WEP, http, Miscellaneous) Einstellungen vornehmen zu können, muss die Automatik zunächst abgeschaltet (Optionsfeld: Automatic deploy...) und durch „Accept“ bestätigt werden. Durch den Knopf „Undo Changes“ werden nach Änderungen in den Unterpunkten diese wieder verworfen und die aktuell eingestellten Parameter vom Server geholt. Der Knopf „Default“ stellt für alle Parameter Standardwerte ein. Der Knopf „Accept“ übergibt die Einstellungen an den Server.

Durch Aktivieren der Automatik werden bei allen CallButtons die markierten Parameter eingestellt, sobald sie Verbindung mit *Rabus Server* aufnehmen.

Im Feld „Configuration Version“ wird die aktuelle Konfigurations-ID angegeben. Sie wird durch Rabus Server bei einer Änderung an den Parametern automatisch erhöht und dient bei einer

CallButton-Kommunikation zum Vergleich mit den Parametern auf dem CallButton. Nur bei einer Abweichung werden Parameteränderungen zum Gerät gesendet.

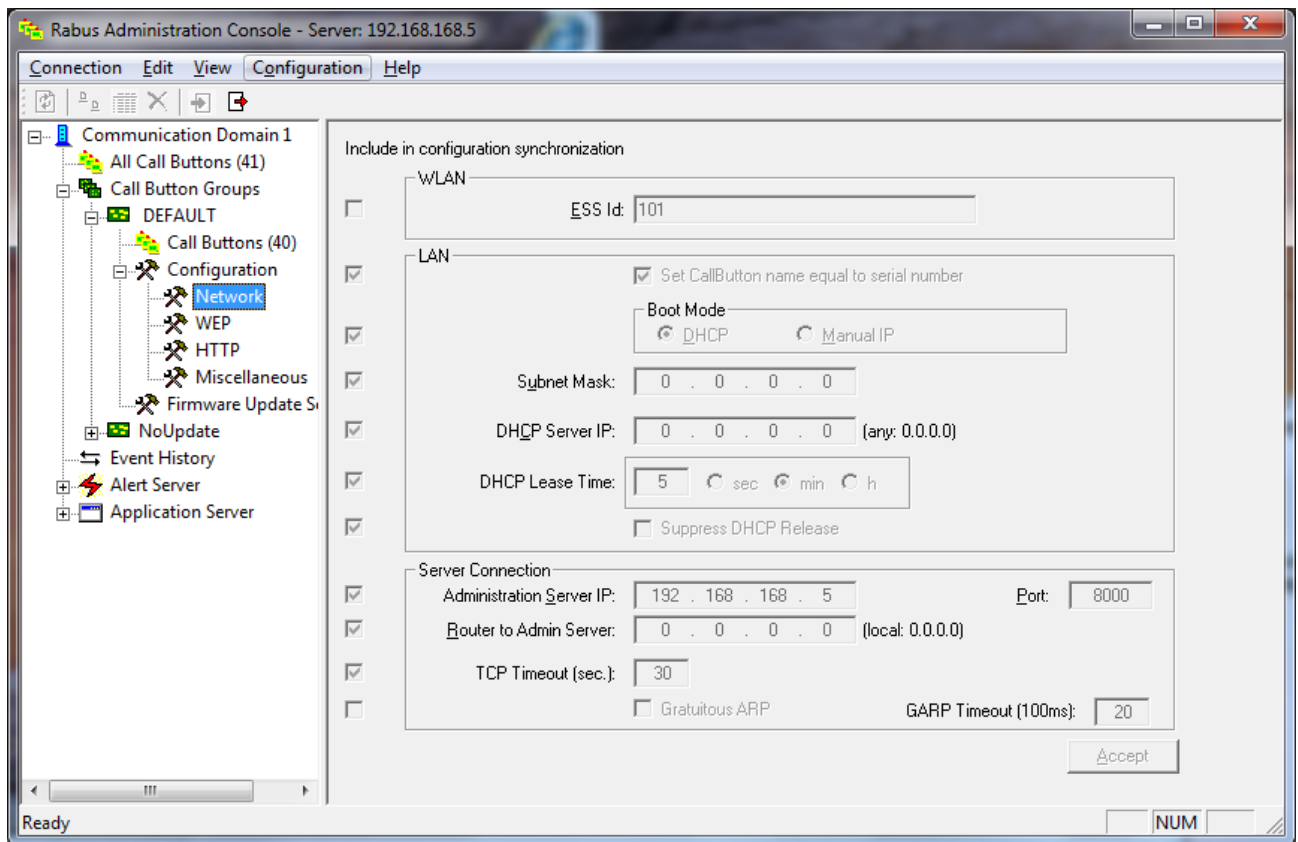
In den Unterpunkten der Komponente **Call Button Config** werden die zu ändernden Parameter definiert. Ein Häkchen im Optionsfeld am linken Rand der jeweiligen Ansicht auf der rechten Seite wird bestimmt, welche Parameter bei einer Änderung berücksichtigt werden sollen.

Lassen Sie bitte äußerste Vorsicht bei der Angabe der zu ändernden Parameter walten. Ein Verlust der Netzwerkverbindung durch falsche Werte kann nur durch Aufsuchen jedes einzelnen CallButtons und Umkonfiguration mit *FBConfig* behoben werden. Bitte beachten Sie Kapitel zur Globalen CallButton-Konfiguration weiter oben.

Eine Beschreibung zu den einzelnen Werten finden Sie in der Dokumentation zu *FBConfig*.

5.8.1 Ansicht Network

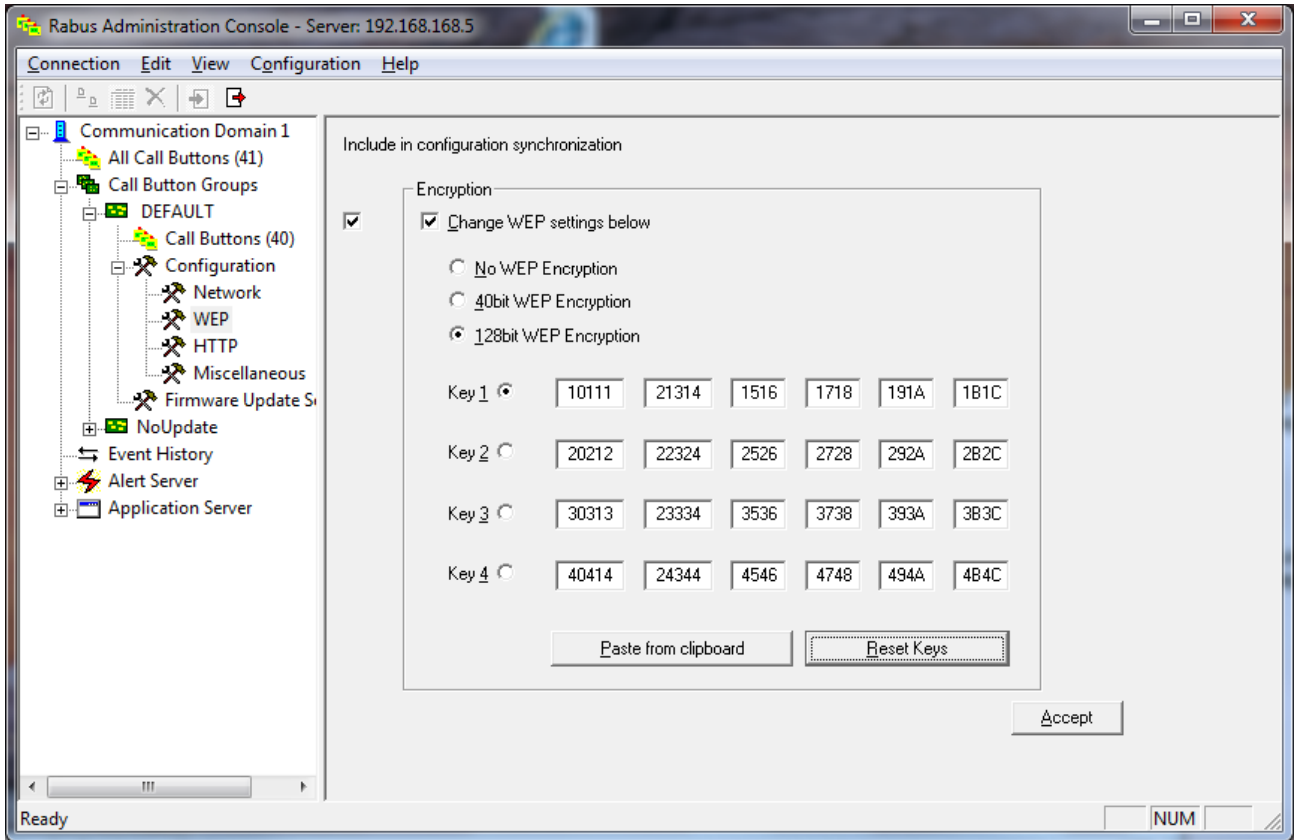
In der Ansicht **Network** werden die wesentlichen Netzwerkparameter eingestellt.



Die markierten Parameter werden global bei allen CallButtons geändert. Im Beispiel in der Abbildung werden nur die Einstellungen zum Webserver verändert.

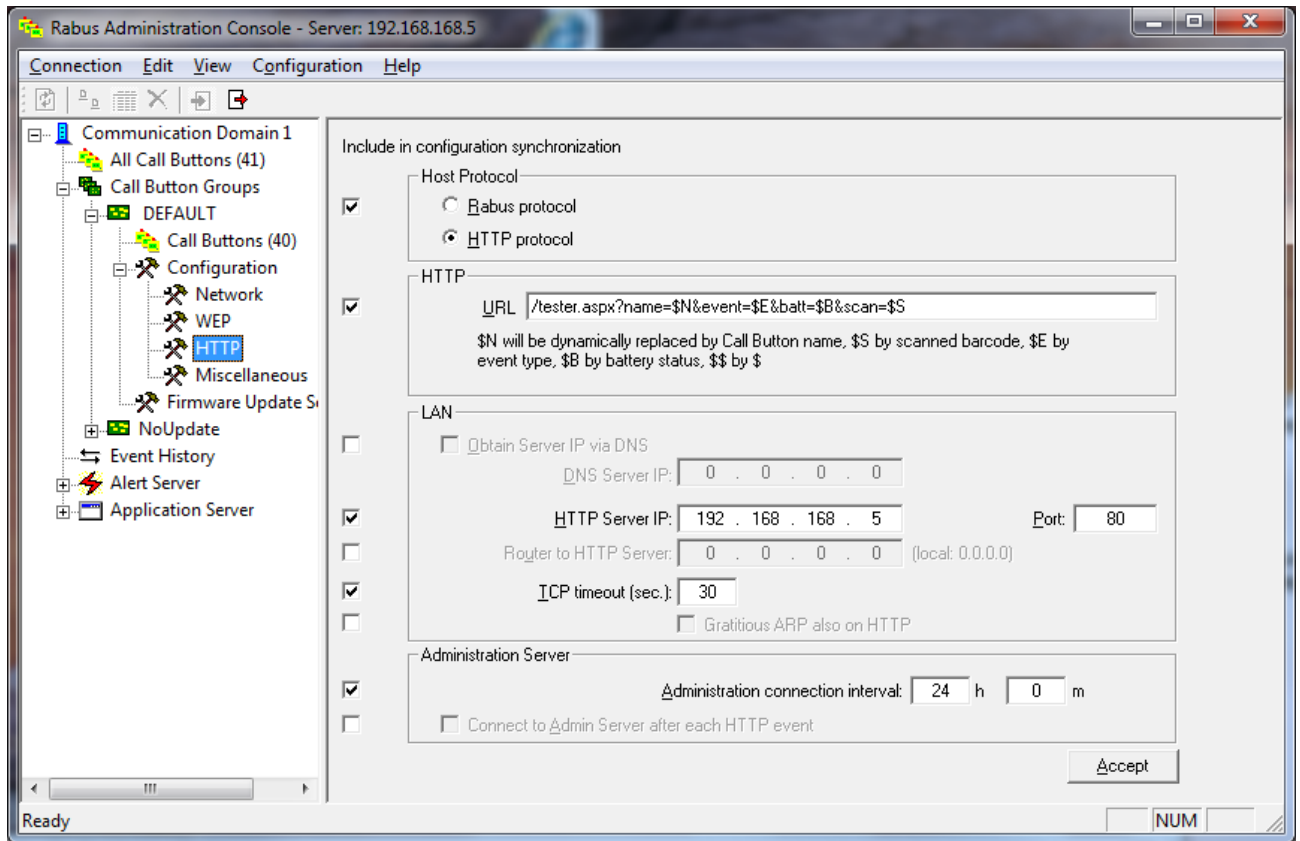
5.8.2 Ansicht WEP

In der Ansicht **WEP** werden die Verschlüsselungsparameter eingestellt.



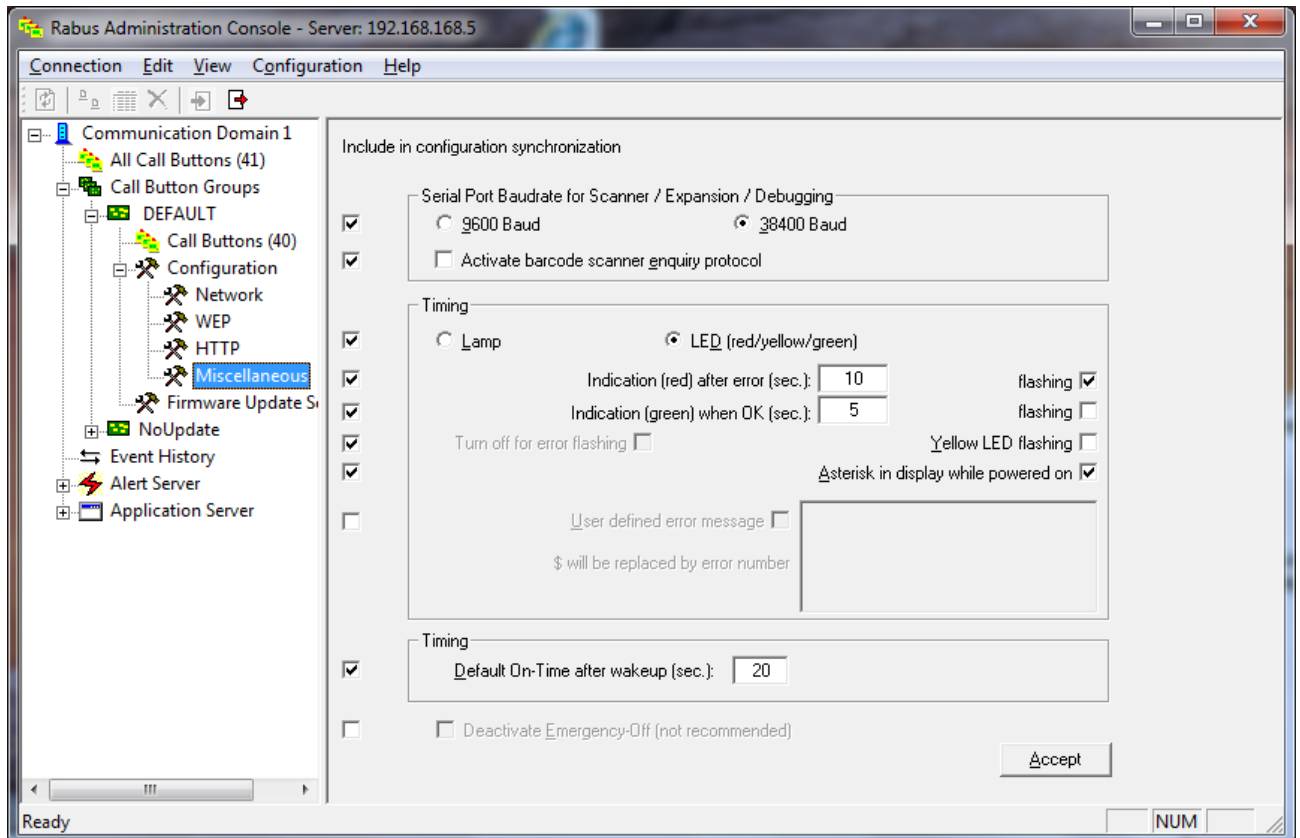
5.8.3 Ansicht HTTP

In der Ansicht **HTTP** werden die Parameter zum HTTP-Protokoll eingestellt.



5.8.4 Ansicht Miscellaneous

In der Ansicht **Miscellaneous** werden diverse andere Parameter eingestellt.

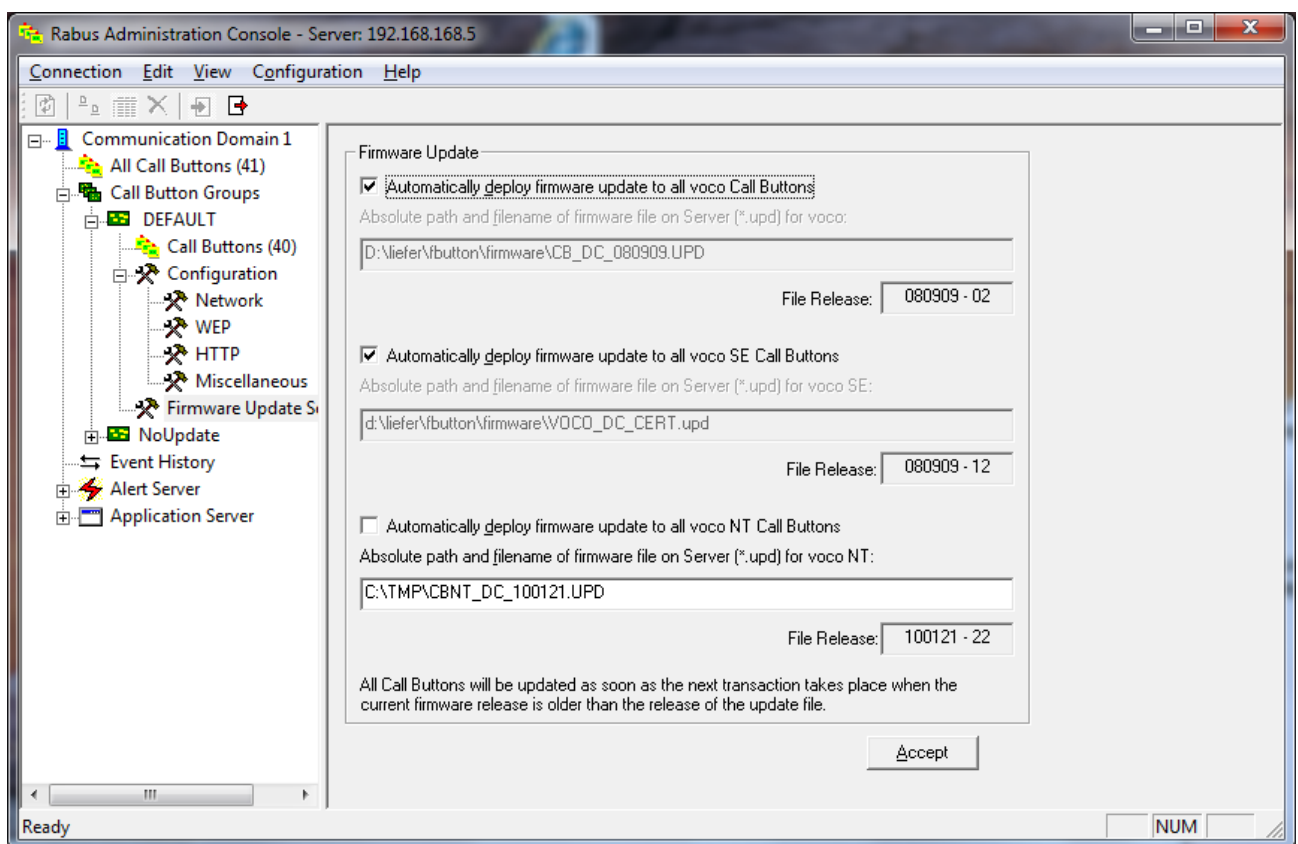


5.9 Ansicht Firmware Update Settings

In der Ansicht **Firmware Update Settings** wird die Überwachung und Verteilung von Firmware-Updates eingestellt. Bei jeder Kommunikation übermittelt ein CallButton seine aktuelle Firmware-Version. Wenn sich die Firmware-Version eines CallButtons von der eingestellten unterscheidet, wird ein Update über WLAN durchgeführt.

Die Dateispezifikation muss mit Laufwerksangabe und absolutem Pfad (oder UNC-Pfad) aus Sicht des Servers angegeben werden.

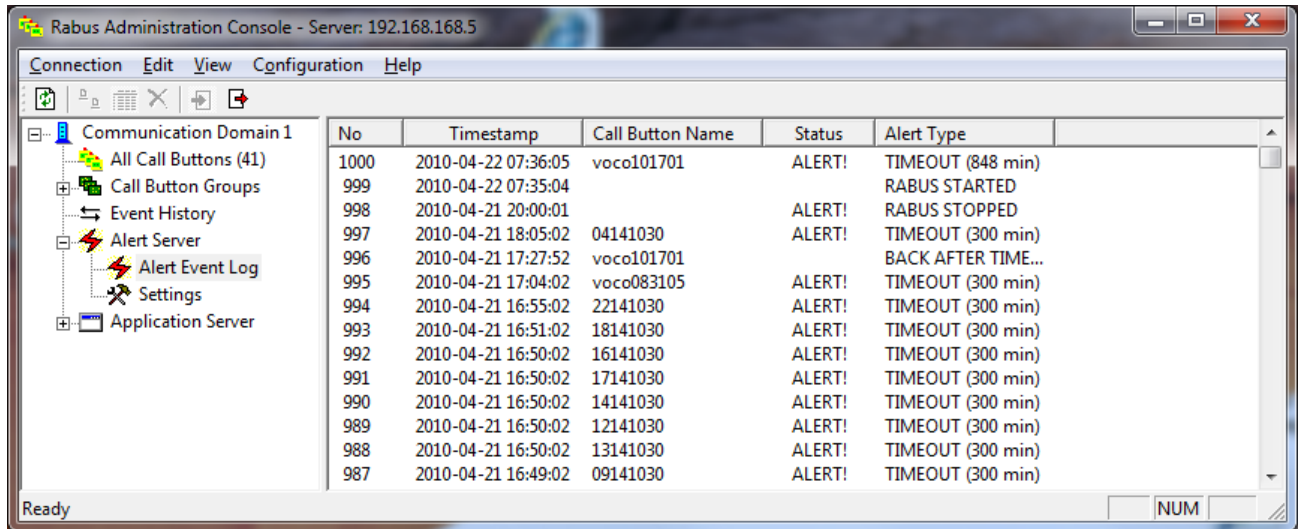
Durch den Knopf „Accept“ werden die Einstellungen auf dem Server aktiviert.



5.10 Komponente Alert Server

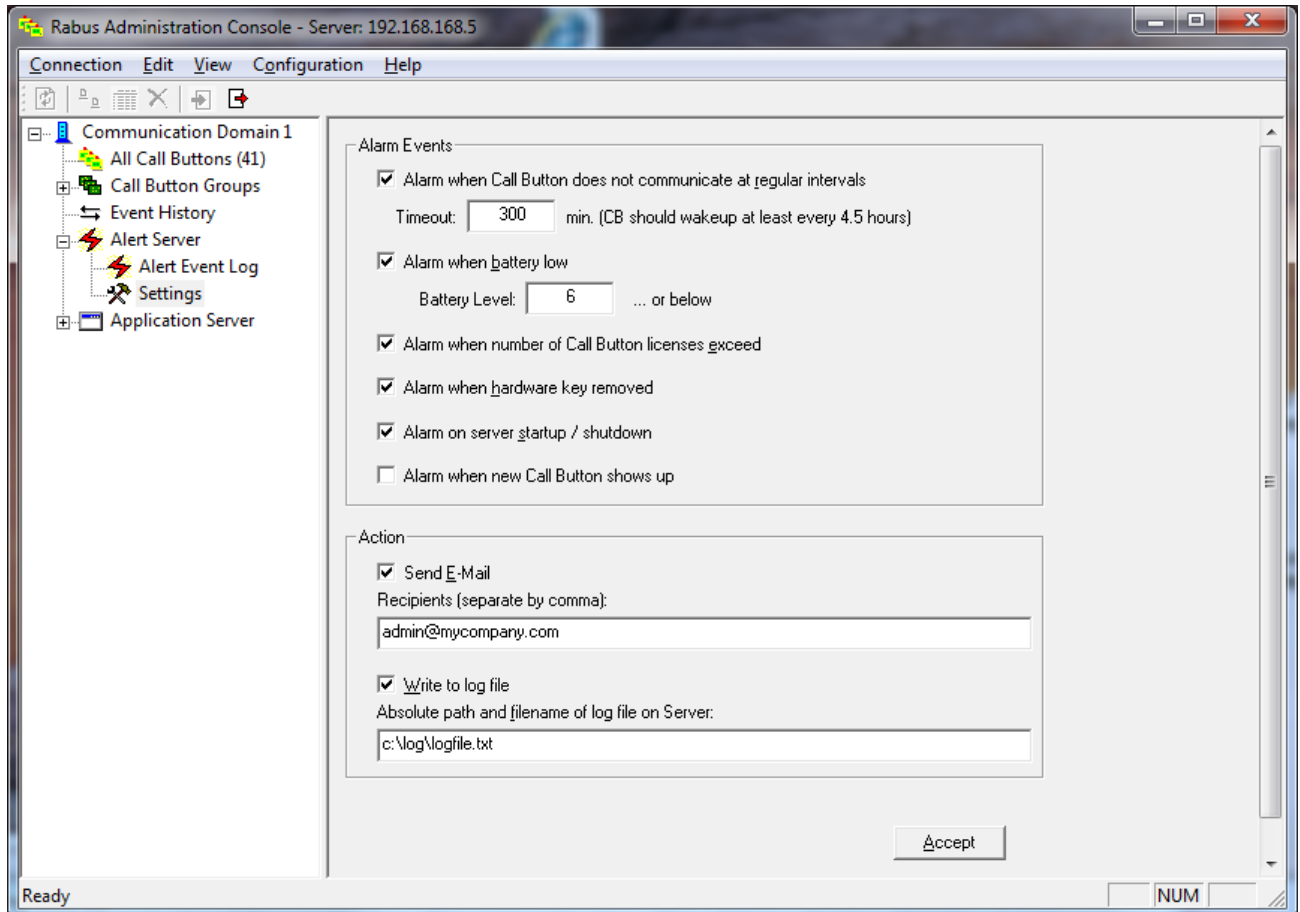
5.10.1 Ansicht Alert Event Log

In der Ansicht **Alert Event Log** wird auf der rechten Seite eine Liste der letzten 200 Alarm-Ereignisse angezeigt.



5.10.2 Ansicht Alert Server Settings

In der Ansicht **Alert Server Settings** können auf der rechten Seite die Alarm-Einstellungen vorgenommen werden.



Im oberen Teil werden die zu meldenden Ereignisse ausgewählt:

- Kommunikations-Timeout

Ein Alarm wird ausgelöst, wenn sich ein CallButton für die eingestellte Zeit in Minuten keine Verbindung aufgenommen hat. Da sich jeder CallButton mindestens alle 255 Minuten (4,5 Stunden) bei Rabus Server meldet, kann mit diesem Alarm festgestellt werden, wenn ein Gerät nicht mehr funktioniert. Ein Wert von z.B. 300 Minuten wäre ein guter Timeoutwert. Ein gegenteiliger Alarm wird erzeugt, wenn sich das betreffende Gerät wieder zurückmeldet.

- Batterie leer

Ein Alarm wird für einen CallButton ausgelöst, wenn die Batterieladung den eingestellten Wert unterschreitet. Ein gegenteiliger Alarm wird erzeugt, wenn die Ladung wieder um 2 Stellen über dem eingestellten Wert ist.

- Lizenz überschritten

Ein Alarm wird ausgelöst, wenn ein CallButton sich versucht zu verbinden und die Anzahl der aktivierten Client-Lizenzen bereits erreicht ist.

- Hardware-Key entfernt

Ein Alarm wird ausgelöst, wenn der Hardware-Key entfernt wurde.

- Server startup / shutdown

Ein Alarm wird jeweils bei Start und Stop des *Rabus Servers* ausgelöst.

Alle Alarm-Ereignisse werden durch Rabus Server gespeichert und können mit *RabusAdmin* durch die Ansicht Alert Log Event angezeigt werden.

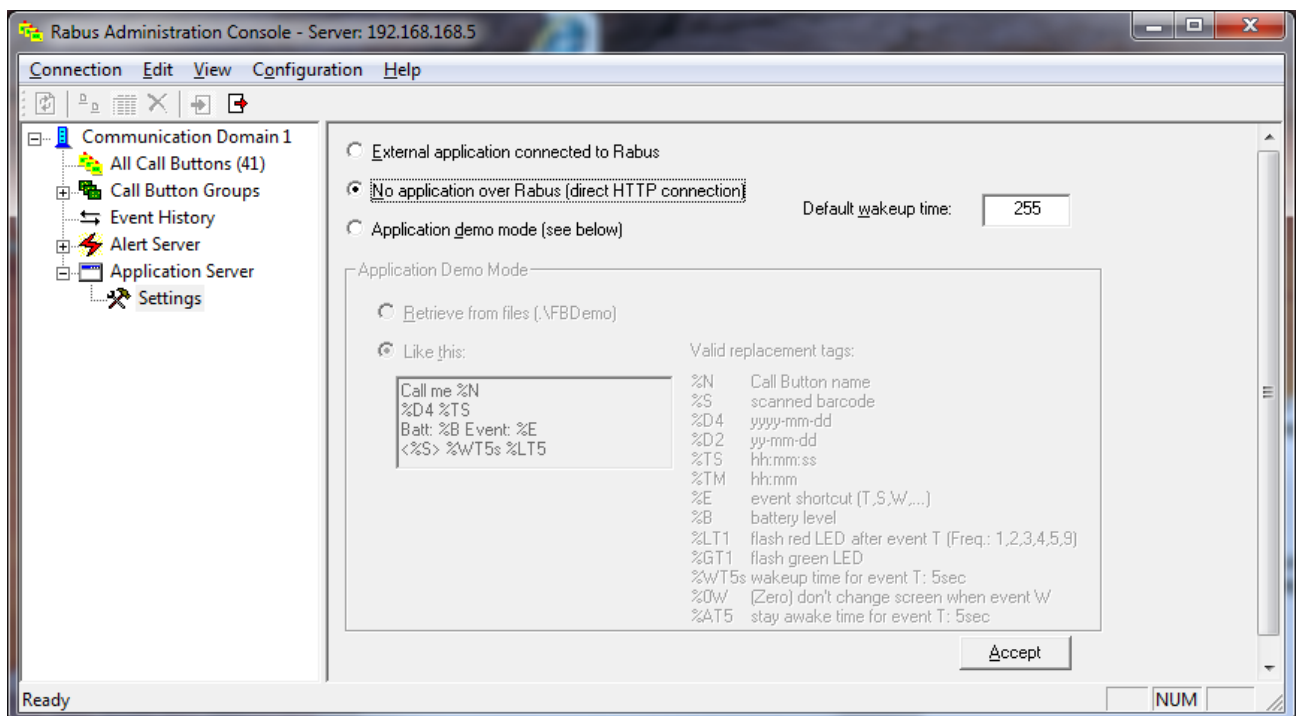
Wenn eine oder mehrere durch Komma getrennte E-Mail-Adressen angegeben werden, wird eine Benachrichtigung für jeden der angegebenen Alarm-Ereignisse an die Empfänger gesendet.

Durch Angabe einer Logdatei (mit Pfadnamen aus Sicht des *Rabus Servers*) werden die Alarm-Ereignisse an diese Datei angefügt.

Durch den Knopf „Accept“ werden die Einstellungen auf dem Server aktiviert.

5.11 Applikations-Demomodus

In der Ansicht **Application Server Settings** können auf der rechten Seite die Einstellungen zur Art der Applikation vorgenommen.



Wenn die obere Option (externe Applikation) aktiviert ist, müssen die Anzeigedaten für die CallButtons durch eine verbundene Applikation generiert werden (*Rabus* als Kommunikationsserver).

Im Falle der zweiten Option (direkte HTTP-Verbindung), wird *Rabus* als reiner Administrationsserver verwendet und sendet keine Anzeigeinformationen zum CallButton. Die Applikation ist in diesem Fall durch den Webserver realisiert.

Bei der dritten Option generiert Rabus als Applikations-Demoserver selbst die Anzeigeinformationen.

Nur wenn die erste Option aktiviert ist, kann sich eine Applikation mit *Rabus Server* verbinden. Bei den beiden anderen Optionen muss noch eine Standard-Weckzeit (zwischen 1 und 255 Minuten) angegeben werden. Dadurch wird sich der CallButton in dieser Zeit nach einem Ereignis mit einem Weckereignis wieder melden. Das Wecken lässt sich nicht deaktivieren. Schließlich müssen wir ja auch jeden Morgen früh aufstehen, wenn der Wecker klingelt ;-)

Wenn beim Applikations-Demoserver die Option **Retrieve from files** aktiviert ist, liest *Rabus Server* entsprechend dem Ereignis eine Datei ein und übergibt den Inhalt als Anzeigeinformation. Das Verzeichnis der Dateien ist durch die INI-Datei (Parameter DemoDir) definiert. Folgende Dateiname wird gelesen:

- <CallButton Name>.<Ereignis-Buchstabe>, wenn vorhanden – sonst:
- <CallButton Name>.txt, wenn vorhanden – sonst:
- FBDemo.< Ereignis-Buchstabe>, wenn vorhanden – sonst:
- FBDemo.txt

Der Ereignis-Buchstabe ist ‘T’ für ein Knopfereignis, ‘S’ für ein Scanereignis, ‘W’ für ein Weckereignis, ‘A’ für ein externes Ereignis.

Wenn eine Datei gefunden wurde, wird sie eingelesen und die Platzhalter werden dynamisch durch folgende Werte ersetzt:

- %N – wird ersetzt durch den Namen des CallButtons
- %S – wird ersetzt durch den gescannten barcode, wenn vorhanden
- %D4 – wird ersetzt durch das Serverdatum (Format: JJJJ-MM-TT)
- %D2 – wird ersetzt durch das Serverdatum (Format: JJ-MM-TT)
- %TS – wird ersetzt durch die Server-Uhrzeit (Format: HH:MM:SS)
- %TM – wird ersetzt durch die Server-Uhrzeit (Format: HH:MM)
- %E – wird ersetzt durch den Ereignis-Buchstaben (s.o.)
- %B – wird ersetzt durch den Batterie-Ladestand (0-9)
- %L[<Ereignis>]<Freq.> – wird nicht angezeigt, sondern bewirkt, dass die Lampe eingeschaltet wird (Blinkfrequenz: <freq>: 1=0.5Hz; 2=1Hz; 3=2Hz; 4=3Hz; 5=5Hz; 9=konstant an). Wird ein Ereignisbuchstabe angegeben, wird die Lampe nur bei diesem Ereignis eingeschaltet (T=Taste gedrückt, A=Externes Ereignis; S=Barcode gescannt). Beispiel: %LT1 wird Lampe

nach jedem Tastendruck mit 0.5Hz bis zum nächsten Ereignis blinken lassen. Bitte beachten Sie den Stromverbrauch der Lampe!

- %W<Ereignis><Zeit> – wird nicht angezeigt, sondern bewirkt, dass die Weckzeit nach dem angegebenen Ereignis (T=Taste gedrückt, A=Externes Ereignis; S=Barcode gescannt) auf die angegebene Zeit (1 bis 255 Minuten [1...255] oder 5 bis 255 Sekunden [5s...255s]) eingestellt wird (T, A, S). Beispiel: %WT2 wird den betreffenden CallButton 2 Minuten nach einem Tastendruck wecken. Zusammen mit dem Platzhalter %L lässt sich bewirken, dass die Lampe nach einem Ereignis für eine bestimmte Zeit leuchtet. Beispiel: %LT2%WT5s bewirkt, dass die Lampe nach einem Tastendruck für 5 Sekunden mit 1Hz blinkt.
- %A<Ereignis><Zeit> –setzt eine Anbleibzeit von <Zeit> Sekunden nach Ereignis vom Typ <Ereignis>.
- %% – wird ersetzt durch das Prozentzeichen (%)

Durch den Knopf „Accept“ werden die Einstellungen auf dem Server aktiviert.

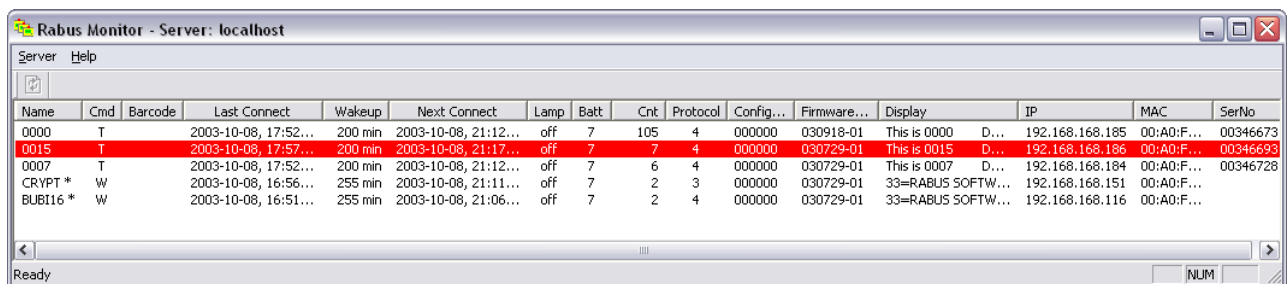
6 RabusMonitor

6.1 Mit Rabus Server verbinden

Beim Start fragt *RabusMonitor* zunächst die Verbindungsangaben zu *Rabus Server* ab (Hostname oder IP-Adresse und TCP-Port, standardmäßig 18301) und baut anschließend die Verbindung auf. Die Angaben werden für zukünftige Aufrufe gespeichert.

Die aktuellen Zustände aller bekannten CallButtons werden angezeigt und kontinuierlich aktualisiert. Jedes eingehende CallButton-Ereignis wird für 10 Sekunden rot unterlegt dargestellt.

Durch Klicken auf die Spaltenüberschrift wird die Liste nach der entsprechenden Spalte sortiert. Durch Ziehen des Spaltenseparators kann die Spalte vergrößert, verkleinert oder versteckt werden.



The screenshot shows a window titled "Rabus Monitor - Server: localhost". It contains a table with the following columns: Name, Cmd, Barcode, Last Connect, Wakeup, Next Connect, Lamp, Batt, Cnt, Protocol, Config..., Firmware..., Display, IP, MAC, and SerNo. The table contains five rows of data, with the second row (Name: 0015) highlighted in red. The status bar at the bottom shows "Ready" and a "NUM" button.

Name	Cmd	Barcode	Last Connect	Wakeup	Next Connect	Lamp	Batt	Cnt	Protocol	Config...	Firmware...	Display	IP	MAC	SerNo
0000	T		2003-10-08, 17:52...	200 min	2003-10-08, 21:12...	off	7	105	4	000000	030918-01	This is 0000 D...	192.168.168.185	00:A0:F...	00346673
0015	T		2003-10-08, 17:57...	200 min	2003-10-08, 21:17...	off	7	7	4	000000	030729-01	This is 0015 D...	192.168.168.186	00:A0:F...	00346693
0007	T		2003-10-08, 17:52...	200 min	2003-10-08, 21:12...	off	7	6	4	000000	030729-01	This is 0007 D...	192.168.168.184	00:A0:F...	00346728
CRYPT *	W		2003-10-08, 16:56...	255 min	2003-10-08, 21:11...	off	7	2	3	000000	030729-01	33=RABUS SOFTW...	192.168.168.151	00:A0:F...	
BUB116 *	W		2003-10-08, 16:51...	255 min	2003-10-08, 21:06...	off	7	2	4	000000	030729-01	33=RABUS SOFTW...	192.168.168.116	00:A0:F...	

Angezeigt werden folgende Informationen über die CallButtons:

- CallButton-Name
- Letztes Ereignis (T–Tastendruck, A–Externes Ereignis, S–Barcode gescannt, W–Weckereignis, C–Konfigurationsende, U–Firmwareupdate-Ende)
- Barcode wenn das letzte Ereignis ein Scanvorgang war
- Zeitpunkt des letzten Ereignisses
- Nächste Weckzeit
- Nächstes erwartetes Weckereignis
- Aktueller Lampenzustand (Blinkfrequenz)
- Batterieladung (0 to 9)
- Anzahl der Ereignisse seit Rabus Server gestartet wurde
- CallButton Protokolltyp
- Aktuelle Konfigurationsversion, wenn der Automatische Konfigurationsupdate aktiviert ist und sich die Version des CallButtons von der aktuell konfigurierten unterscheidet, so wird dahinter ein Ausrufungszeichen in Klammern angezeigt
- Pluszeichen, wenn der CallButton beim Automatischen Konfigurationsupdate berücksichtigt wird
- Aktuelle Firmwareversion, wenn der Automatische Firmwareupdate aktiviert ist und sich die Version des CallButtons von der aktuell konfigurierten unterscheidet, so wird dahinter ein Ausrufungszeichen in Klammern angezeigt

- Pluszeichen, wenn der CallButton beim Automatischen Firmwareupdate berücksichtigt wird
- Aktueller Anzeigeinhalt
- IP-Adresse
- MAC-Adresse
- Seriennummer