

BS1000 messenger to web server

Einführung

Die BS1000 LAN Basisstation für das Arexx-Multilogger System stellt einen Messenger-Dienst zur Verfügung, womit man von der Basisstation http-Kommandos an externen Web-Servern abwickeln kann. Die http-Kommandos ermöglichen den Datentransfer der Messdaten in eine Datenbank mittels Script-basierter Webdiensten wie MySQL/Apache oder Microsoft SQLServer/asp.net.

Für jeden eintreffenden Messwert überprüft der Messenger zuvor festgelegte Regeln. Eine Regel ist ein Auftrag, der ausgeführt wird, sobald eine zugehörige Bedingung zutrifft.

Regeln werden in einem Rule Editor festgelegt, und die damit erstellte Regel-Datei (oder "Rule-Datei") wird anschließend zum BS1000 übertragen, in dem die Regelsammlung sofort nach Eintreffen aktiviert wird. Zu den in den Aufträgen definierten Kommandos gehören Email-Nachrichten, HTTP-Kommandos oder die Aktivierung des eingebauten Summers. In diesem Kapitel werden wir die HTTP-Kommandos behandeln.

HTTP-Kommandos

Ein HTTP-Kommando (Englisch: HTTP-Request) enthält folgende Informationen:

-Kommandotyp: POST oder GET

-URL: Die URL-Adresse des Webservices, wobei man eine Port-Nummer zur URL angeben kann. Die Port-Nummer muss von der URL durch einen Doppelpunkt getrennt werden.

-Datensatz: Ein benutzerdefinierter Datensatz, der die zu übertragende Information als String enthält. Dieser String wird codiert in base64.

Falls das HTTP-Kommando vom POST-Typ ist, wird der Datensatz dem HTTP-Kommando angehängt. Falls das HTTP-Kommando vom GET-Typ ist, wird der Datensatz dem HTTP-Kommando als String angehängt, getrennt von einem '?'-Fragezeichen. Auf der Serverseite legt der gewählte Kommandotyp (POST oder GET) fest, wie die Daten verarbeitet werden sollen.

Das BS1000-System setzt den Datensatz zum Transfer der vorliegenden Messdaten zusammen. Variablen (die mit einem '\$'-Zeichen beginnen) werden durch aktuelle Messdaten ersetzt, wie zum Beispiel Messwert, Sensornummer, usw. Anschließend wird der zusammengesetzte String mittels base64-Code codiert und als HTTP-Kommando an den Webserver abgesetzt. Dabei stehen folgende Variablen zur Verfügung:

Variable	Umschreibung
\$v	Messwert
\$q	Sensortyp 1 = Temperatur (°C), 3 = RH% (%), 5=CO2 (ppm)
\$i	Identifikationsnummer des Sensors
\$r	rssi--Wert (Signalpegel in dBm)
\$h	Stundenangabe der Uhrzeit der Messung
\$m	Minutenangabe der Uhrzeit der Messung
\$s	Sekundenangabe der Uhrzeit der Messung
\$Y	Jahresangabe der Uhrzeit der Messung
\$M	Monatsangabe der Uhrzeit der Messung
\$D	Tagesangabe der Uhrzeit der Messung
\$S	Zeitangabe der Messung in Sekunden seit 1-1-2000 UTC
\$w	„Fehlt“-Parameter, der den Zeitpunkt der letzten Übertragung eines Messwertes an den http-Server dokumentiert. Dieser wird zur Synchronisation des Temp-Loggers benötigt.
\$t	<i>time string</i> ; Zeitpunkt der Messung im Format: hh:mm:ss
\$d	<i>datum string</i> ; Datum der Messung im kurzen Datumsformat

Abgesehen von \$w und \$\$ gilt für alle Zeitangaben UTC plus Zeitzone-Offset (laut Definition in der *Time server* – Konfigurationsseite. Für die Zeitangaben \$w und \$\$ gilt die UTC.

Das HTTP-Kommando wird in base64-Format kodiert. Dabei werden nicht-alphanumerische Zeichen in %hh-strings umgewandelt, wobei „hh“ eine hexadezimale Zahl darstellt. Eine Ausnahme bilden dabei die Reihen '&&' und '==': diese werden umgesetzt in '&', beziehungsweise '='. Die Nachricht zum HTTP-Request wird im Request-Header (die Kopfzeile POST) weitergereicht, beziehungsweise an die URL des Requests (GET) hinzugefügt. Dabei wird das Trennzeichen '?' zwischen URL und Nachricht hinzugefügt.

Beispiel einer Nachricht:

id==\$i&&value==\$v

In diesem Beispiel wird ein Webserver programmiert, um den angegebenen String in zwei Parametern 'id' und 'value' zu dekodieren. Mit dieser Methode kann die BS1000 ohne zugeschalteten PC eine Webseite mit aktuellen Messdaten versorgen.

Die Serverseite

Normalerweise werden die HTTP-Kommandos an eine speziellen Webseite adressiert, die über Script-Werkzeuge verfügt. Eine solche Webseite könnte www.server.com/multilogger.php heißen. Diese Webseite würde dann Scripts bereitstellen, welche die Daten decodieren, deren Inhalt überprüfen und die Messdaten in ein Speichermedium übertragen. Andere Webseiten könnten dann zur Darstellung der Messdaten verwendet werden .

Dieses Dokument ist kein Handbuch zur Script-Erstellung; wir verweisen dazu auf die vielfältige Hilfsdokumentation an anderer Stelle. Sie können dazu zum Beispiel die PHP-Hilfe zum Thema '*variables from outside PHP*' studieren. Um Ihnen den schnellen Einstieg zu ermöglichen, werden wir im nachfolgenden Kapitel zeigen, wie man Variablen auf einer PHP-Seite verarbeiten kann:

```
<?php
// multilogger.php
// benötigt 6 Variablen, getrennt mittels '&'-Zeichen:
// zum Beispiel: abcdef&&$d&&$t&&$i&&$v

// Argument 0 = 'password' (abcdef)
// Argument 1 = $d Datum
// Argument 2 = $t Zeit
// Argument 3 = $i Sensor-Identifizier
// Argument 4 = $v Sensor Wert

$args = explode("&", $QUERY_STRING );
$nargs = count($args);

if ($nargs != 5)
{
    die();
}

if ($args[0] != "abcdef")
{
    die();
}

$date = urldecode($args[1]) ;
$time = urldecode($args[2]) ;
$device = urldecode($args[3]);
$temperature = urldecode($args[4]);

$date = str_replace(""," ", $date);
$time = str_replace(""," ", $time);
$device = str_replace(""," ", $device);
$temperature = str_replace(""," ", $temperature);
// Daten Speichern
$db = mysql_connect('server', 'user', 'password');
$result = mysql_select_db('database_name', $db);

$result = mysql_query("delete from temperature where (device ='$device')");
$result = mysql_query("INSERT INTO temperature (logdate, logtime, device, temperature)
VALUES ('$date', '$time', '$device', '$temperature' )", $db);

$result = mysql_close($db);
?>
```

Beispiel einer PHP-Seite

In diesem Fall werden Variablen ohne Variablennamen benutzt. Das bedeutet, dass die Kommandozeile des BS1000-Systems die benötigten Variablen genau in der richtigen Reihenfolge anliefern muss. Die korrekte Kommandostruktur muss deshalb folgendermaßen aussehen:

```
abcdef&&$d&&$t&&$i&&$v
```

Die Variablenstruktur wird in ein String-Array (\$args) zerlegt. Die Variablenzahl muss in diesem Fall 5 sein und die erste Variable ist das Passwort. Die Variablenzeile wird codiert in base64-Format. Zur Vermeidung einer sog. *SQL-Einschleusung* (Siehe dazu den Wikipedia-Eintrag: [SQL-Injection](#)) verwenden wir einen einfachen Zeichentausch. Wir zeigen diese Details an dieser Stelle, um Sie an die Vorsichtsmaßnahmen gegen Angriffe auf die Datenbank zu erinnern. Auch sollte man die Datenstruktur für Datum und Uhrzeit kontrollieren, aber das wurde im Beispiel nicht aufgenommen. Da das BS1000-System nur aktuell gültige Daten versendet, kann man Daten mit abweichenden Datumsangaben ablehnen. Die letzte Aktion ist der Speichervorgang, wobei die Messdaten mit einem SQL-Insert-Kommando in die Datenbank übertragen werden.