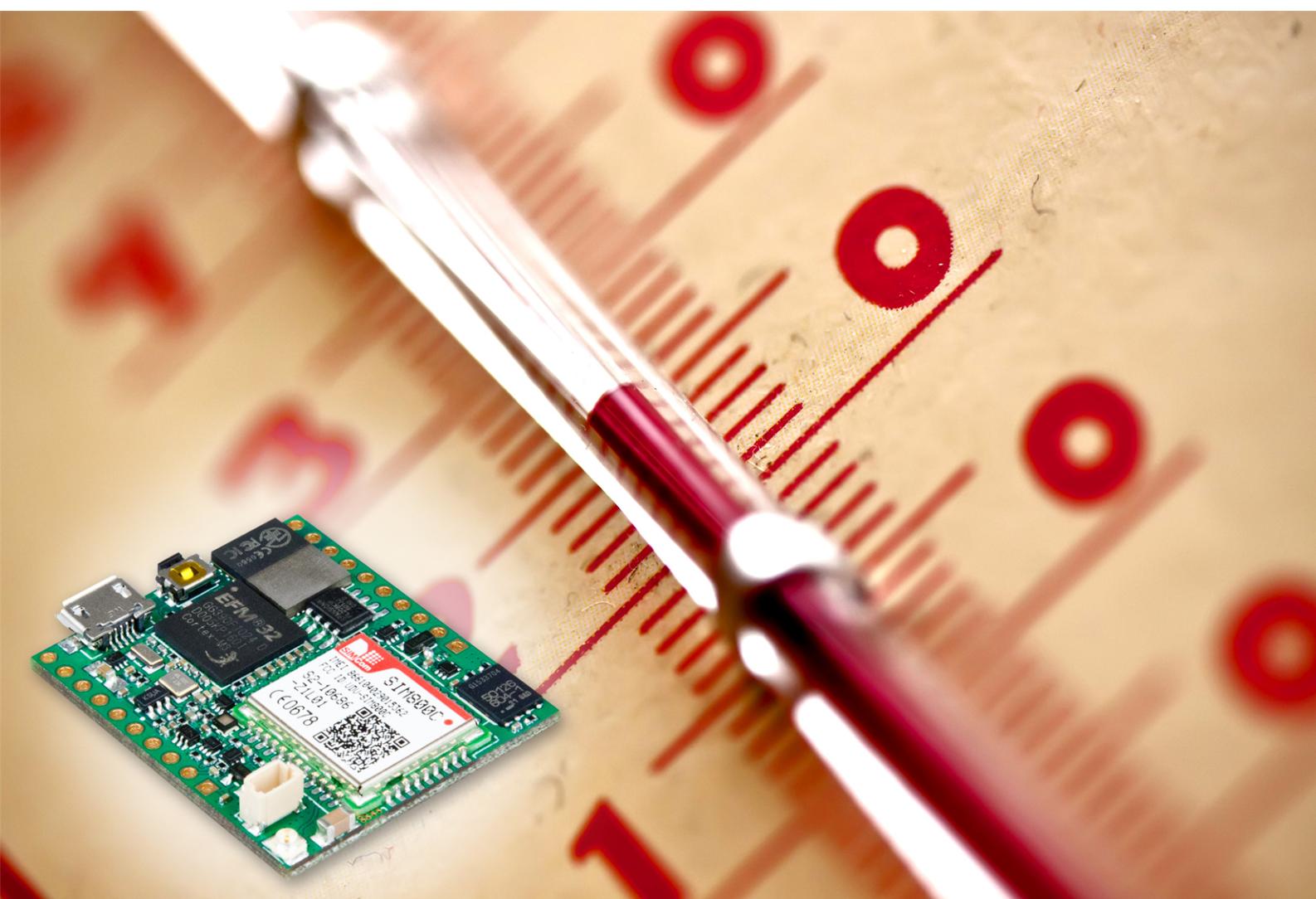


Vom Sensorsignal zur Visualisierung im Web

Anwendung für das Internet der Dinge rasch realisieren



Tutorial

Das folgende Tutorial demonstriert Ihnen anhand des Beispiels eines "Wireless Thermometers" wie Sie mittels des rapidM2M M3 eine Anwendung für das Internet der Dinge rasch realisieren. Dieses Tutorial ist in 5 Phasen unterteilt:

- Phase 1: Festlegen des Connectors
- Phase 2: Erstellen des Applikations-Scripts
- Phase 3: Visualisieren der Daten am Cloud-Server
- Phase 4: Zugriff auf die Daten und Konfigurationen über die REST-API des Cloud-Servers mittels rapidM2M Playground
- Phase 5: Erstellen eines eigenen Dashboards

Sie erfahren wie Sie die für den Datenaustausch zwischen rapidM2M M3 , Cloud-Server und externen Systemen erforderliche Datenstruktur (d.h. den Connector) festlegen, ein Applikations-Script mit Hilfe des rapidM2M Toolset erstellen und dieses per USB-Schnittstelle in das rapidM2M M3 laden. Weiters erlernen Sie die Verwendung der Auswertungen zur Darstellung der Daten am Cloud-Server und erhalten einen Überblick wie mit Hilfe des rapidM2M Playground Daten und Konfigurationen über die REST-API des Cloud-Servers ausgelesen werden können. Abschließend erarbeiten Sie ein eigenes Dashboard zur Darstellung der Daten und Eingabe eines Konfigurationsparameters wobei dieses von Ihnen geschaffene Dashboard über die REST-API auf die entsprechenden am Cloud-Server gespeicherten Informationen zugreift.

1.1 Voraussetzungen

Für dieses Tutorial benötigen Sie folgende Komponenten:

- PC mit Microsoft Betriebssystem und aktuellem Browser (Empfehlung: Google Chrome, Mozilla Firefox)
- rapidM2M M3 (300443)
- USB-Kabel A-Stecker auf Micro-B-Stecker
- rapidM2M Toolset (Mindestversion 1.18.4, Download unter: www.microtronics.at/Toolset)
- Account am Cloud-Server
- Ergänzende Dateien zu diesem Tutorial (Downloadbar unter: http://support.microtronics.at/tutorial/Tutorial_wireless_thermometer_additional_files.zip)

1.2 Detaillierte Beschreibung

In Phase 1 wird eine neue Messstelle für den Betrieb des "Wireless Thermometers" angelegt sowie der erforderliche Connector definiert. Applikations-Scripts für das rapidM2M M3 können über eine nahezu beliebige Anzahl an Konfigurationsparametern und Messkanälen verfügen. Auch deren Datentyp kann innerhalb gewisser Grenzen frei gewählt werden. Der Connector dient dazu, dem Cloud-Server diesen Aufbau mitzuteilen, um die Messdaten und Konfigurationen in Verbindung mit der Oberfläche des Servers nutzen zu können (Auswertungen, Visualisierungen, Grafiken, etc.). Auch um den Zugriff auf die Messdaten und Konfigurationen über die API zu ermöglichen, muss dem Cloud-Server dieser Aufbau bekannt sein. Es empfiehlt sich bei der Entwicklung einer M2M / IoT Anwendung mit der Definition des Connectors zu beginnen. Um eine rasche Reaktionszeit zu erreichen, wird abschließend die Verbindungsart auf „online“ und das Aufzeichnungsintervall auf 10sec. geändert.

Phase 2 demonstriert anhand des Beispiels eines "Wireless Thermometers" wie Sie mit Hilfe des rapidM2M Toolset ein Applikations-Script für das rapidM2M M3 erstellen und über die USB-Schnittstelle in das Gerät laden.

In Phase 3 erfolgt die Konfiguration einer Auswertung, um die aktuellsten Messwerte sowie den Verlauf der Messwerte während der letzten 30min. über die Oberfläche des Cloud-Servers anzuzeigen. Gemäß der in Phase 1 festgelegten Definition des Connectors werden durch das Applikations-Script die Messwerte Batteriespannung, USB-Ladespannung und Temperatur bereitgestellt.

Die optionale Phase 4 veranschaulicht wie mittels des rapidM2M Playground über die REST-API des Cloud-Servers auf Messdaten und Konfigurationen zugegriffen werden kann. Sie dient somit dazu sich mit den bereitgestellten Ressourcen und deren Handhabung vertraut zu machen. Der erste Teil dieser Phase demonstriert wie die aktuellsten Messwerte der Batteriespannung, USB-Ladespannung und Temperatur über die API vom Server gelesen werden. In einem zweiten Schritt

wird zunächst die komplette Konfiguration ausgelesen und anschließend ein einzelner Parameter - und zwar die Verbindungsart von "Online" auf "Wakeup" - geändert.

Phase 5 erläutert die Entwicklung eines Dashboards welches per REST-API auf die am Cloud-Server gespeicherten Messdaten und Konfigurationen zugreift. Dieses verfügt über eine Ansicht zur Eingabe der User Credentials sowie einer weiteren für die Anzeige der aktuellsten der durch das Applikations-Script "Wireless Thermometer" bereitgestellten Messwerte und Eingabe des gewünschten Aufzeichnungsintervalls.

1.3 Script "Wireless Thermometer"

Das Script erfasst periodisch Messdaten und überträgt diese an den Cloud-Server. Sowohl das Intervall für die Aufzeichnung als auch die Übertragung der Messdaten lässt sich konfigurieren. Es ist auch möglich, die Verbindungsart zu konfigurieren. Wird die Verbindungsart "online" gewählt, werden die Messdaten unverzüglich sobald sie erstellt wurden zum Server übertragen. Aufgezeichnet werden die Batteriespannung, USB-Ladespannung und die Temperatur. Der aktuelle Betriebszustand wird mittels RGB-LED signalisiert. Bei bestehender Verbindung mit dem Cloud-Server leuchtet die LED blau. Während des Verbindungsaufbaus flackert die LED blau. Schlug der letzte Verbindungsaufbau fehl, blinkt die LED bis zum nächsten Verbindungsaufbauversuch rot. War der letzte Verbindungsaufbau hingegen erfolgreich und wartet das rapidM2M M3 auf die nächste Kontaktaufnahme mit dem Cloud-Server (z.B. im Verbindungsmodus "Intervall"), ist die LED ausgeschaltet. Eine Verbindung zum Server wird entweder automatisch durch das Gerät nach Ablauf des Übertragungsintervalls bzw. Empfang einer Wackup-SMS oder manuell durch Drücken der Taste ausgelöst.

Hinweis: Die im Folgenden angegebenen Einheiten beziehen sich auf den Datenaustausch zwischen rapidM2M M3 und dem Cloud-Server. Mittels des Attributs "vscale" können die Einheiten bei der Definition des Connectors für die Anzeige über die Oberfläche des Cloud-Servers bzw. die Ausgabe über die REST-API angepasst werden. So wird beispielsweise das Übertragungsintervall für die Anzeige/Eingabe von Sekunden auf Minuten umgerechnet.

Das Script erzeugt und überträgt folgende Messwerte:

Messwert	Einheit	Erklärung
Batteriespannung	mV	Spannung, die am Anschluss für den Akku des rapidM2M M3 anliegt Wird das rapidM2M M3 per USB geladen, entspricht der Messwert der Ladespannung. Andernfalls entspricht der Messwert der Spannung des angeschlossenen Akkus.
USB-Ladespannung	mV	Spannung, die am USB-Anschluss des rapidM2M M3 anliegt
Temperatur	0,1 °C	Messwert, der vom Temperatursensor TMP112 (IC5) des rapidM2M M3 gelesen wird

Das Script bietet folgende Konfigurationsmöglichkeiten:

Parameter	Einheit	Erklärung
Übertragungsintervall	sec.	zeitlicher Abstand der Übertragungen
Aufzeichnungsintervall	sec.	zeitlicher Abstand der Messdatenaufzeichnungen
Verbindungsart	---	o: Das Gerät meldet sich im Übertragungsintervall. 1: Das Gerät meldet sich im Übertragungsintervall. Es kann jedoch auch eine Verbindung durch den Server ausgelöst werden. 2: Das Gerät trennt die Verbindung nicht und übermittelt kontinuierlich die Messdaten.

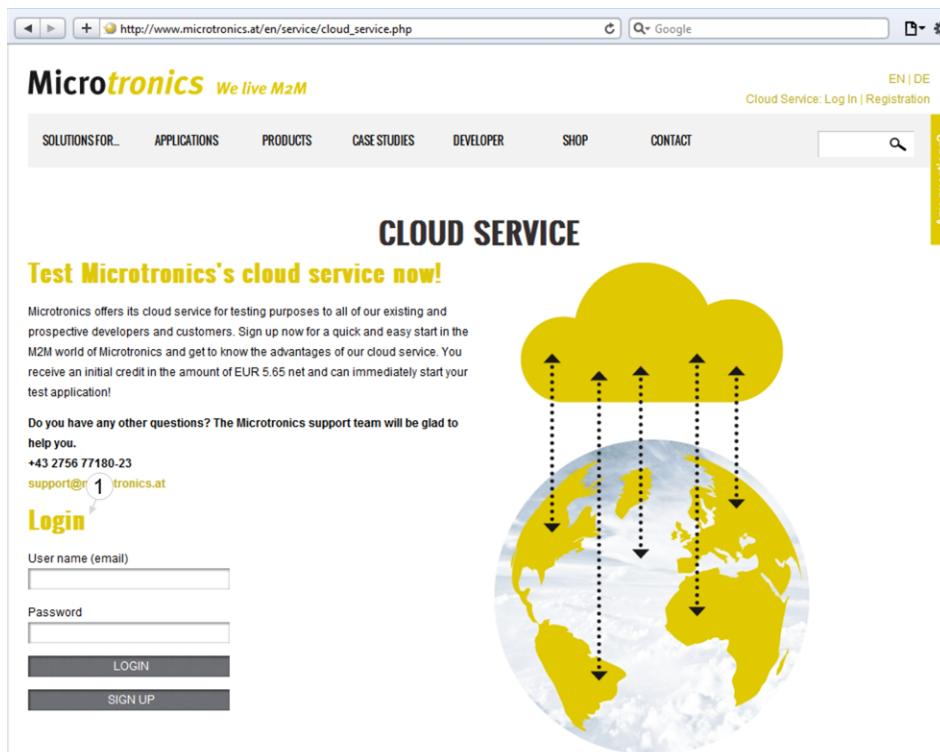
1.4 Dashboard "rapidM2M UAPI SAMPLE"

Das Dashboard verfügt über zwei Ansichten, die im Folgenden als "Login-Page" und "Content-Page" bezeichnet werden. Nach dem Öffnen des Dashboards wird zunächst die "Login-Page", welche die Eingabe der User Credentials ermöglicht, angezeigt. Bei Eingabe gültiger Login-Daten wechselt die Ansicht zur "Content-Page". Neben der Ausgabe der aktuellsten der durch das Applikations-Script bereitgestellten Messwerte für Batteriespannung, USB-Ladespannung und Temperatur kann auch das Aufzeichnungsintervall über die "Content-Page" eingegeben werden. Die Auswahl der Messstelle, deren Informationen dabei angezeigt werden, erfolgt automatisch. Gewählt wird die erste Messstelle des ersten Kunden für den der Benutzer die Zugriffsberechtigung besitzt. Die getroffene Auswahl wird in der "Content-Page" oberhalb des Bereichs für die Messwerte angezeigt. Die Messwerte selbst werden im Sekundenintervall aktualisiert. Mittels des Logout-Buttons kann wieder zur "Login-Page" zurückgewechselt werden wodurch auch die Aktualisierung der Messwerte deaktiviert wird. Beim Wechsel zurück zur "Login-Page" werden die in die Eingabefelder eingegebenen User Credentials nicht gelöscht. Dies hat sich für die Testphase, in der häufig zwischen den beiden Ansichten gewechselt wird, als komfortabel erwiesen. Dieses Verhalten sollte jedoch bei produktiven Dashboards keinesfalls in der Form implementiert werden. Das Dashboard verfügt weiters über einen Bereich zur Anzeige von Statusinformationen (z.B. Fehlercodes), der in beiden Ansichten verfügbar ist. Eine Darstellung der beiden Ansichten sowie eine detaillierte Erläuterung der einzelnen Elemente finden Sie im Kapitel "Phase 5: Erstellen eines eigenen Dashboards" auf Seite 39.

Schritt-für-Schritt-Anleitung

2.1 Phase 1: Festlegen des Connectors

1. Sollten Sie noch nicht über einen Account am Cloud-Server verfügen, können Sie über die Homepage der Firma Microtronics Engineering GmbH (cloud.microtronics.at) gratis einen Account anlegen.
2. Loggen Sie sich am Cloud-Server (cloud.microtronics.at) ein. Der Benutzername ist die E-Mail-Adresse, die Sie beim Anlegen des Accounts angegeben haben.



Übersicht der Homepage

1 Zugang zum Cloud-Server	
----------------------------------	--

3. Sie befinden sich nun im Bereich "Messstellen / Applikationen" des Cloud-Servers.

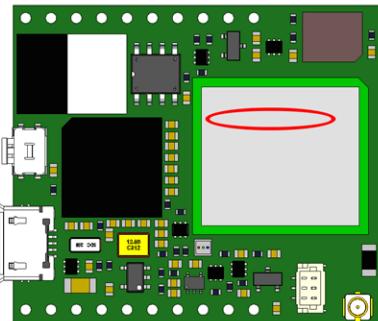
Bereich "Messstellen / Applikationen", in dem alle zur Verfügung stehenden Messstellen angezeigt werden

4. Legen Sie eine Messstelle an.

Anlegen der Messstelle

1 Symbol "Neue Messstelle / Applikation hinzufügen"	3 Eingabefenster für das Anlegen einer neuen Messstelle
2 Feld zur Eingabe der Seriennummer	4 Button "hinzufügen"

Die Seriennummer entspricht der IMEI des rapidM2M M3 ergänzt durch eine abschließende Null. Sie finden die IMEI auf dem Modem des rapidM2M M3 .



rapidM2M M3

Bei der "Applikations-Vorlage" wählen Sie "Standard".

Auswahl der Applikations-Vorlage

1 Dropdown-Liste der verfügbaren Applikations-Vorlagen	3 Eingabefenster zur Auswahl der Applikations-Vorlage
2 Button "hinzufügen"	

5. In der Liste der Messstellen wird nun die neu hinzugefügte Messstelle angezeigt.

Bereich "Messstellen / Applikationen", in dem alle zur Verfügung stehenden Messstellen angezeigt werden

1	öffnet die Eingabemaske zur Konfiguration der Messstelle
2	löscht die Messstelle / Applikation. Das Messgerät wird in den Pool des Kunden verschoben.
3	Seriennummer des mit der Messstelle verknüpften Geräts. Durch Klicken auf die Seriennummer wird die Eingabemaske zur Konfiguration des Geräts geöffnet .
4	Name der Messstelle / Applikation. Durch Klicken auf den Namen wird die Eingabemaske zur Konfiguration der Messstelle / Applikation geöffnet .
5	Liste der Messstellen / Applikationen
6	Status des Messgeräts. Neben dem Statussymbol werden auch, abhängig vom aktuellen Betriebszustand, weitere Symbole eingeblendet.
7	Schaltflächen zum Wechsel zwischen der Anzeige von verbindungsspezifischen und applikationsspezifischen Steuerelementen
8	Verbindungs- bzw. applikationsspezifische Steuerelemente der Messstelle / Applikation

6. Klicken Sie auf den Namen der Messstelle in der Liste der Messstellen / Applikationen, um die Eingabemaske zur Konfiguration der Messstelle zu öffnen.

7. Klicken Sie in der Eingabemaske zur Konfiguration der Messstelle auf "Steuerung", um die Eingabefelder zum Hochladen eines kompilierten Scripts und zur Konfiguration der Datenstruktur (d.h. des Connectors) einzublenden. Setzen Sie die Parameter wie folgt:

The screenshot shows the configuration interface for a measurement station. At the top, there are fields for 'Kunde*' (support@microtronics.at), 'Name*' (8661040290xxxxx), 'Gerät S/N' (8661040290XXXXXX), and 'Applikations-Vorlage' (nicht zugeordnet). Below these, under the 'Steuerung' section, are dropdown menus for 'Script Typ' (set to 'Pawn') and 'Script Quelle' (set to 'Hochladen eines kompilierten Scripts'). A button 'Durchsuchen...' (Browse...) is next to the 'Datei hochladen' field. A scrollable text area labeled 'Datenstruktur' contains the following code:

```

1 //=====
2 //Wireless Thermometer
3 //=====
4 //===
5 //== Settings
6 //===

```

Eingabemaske zur Konfiguration der Messstelle / Applikation, Abschnitt "Steuerung"

1 Konfigurationsabschnitt "Steuerung"

2 Auswahl des Script-Typs

Wählen Sie hier "Pawn".

Hinweis: Erst nach dieser Auswahl wird der Parameter "Script Quelle" eingeblendet.

3 Auswahl, ob das Script direkt am Cloud-Server editiert werden soll oder ein bereits vorab kompiliertes Script verwendet werden soll.

Wählen Sie hier "Hochladen eines kompilierten Scripts".

Hinweis: Erst nach dieser Auswahl wird der Parameter "Datei hochladen" eingeblendet.

4 Button zum Öffnen des Dialogs zur Auswahl des hochzuladenden Script Binary-Files (*.amx)

Geben Sie hier die Datei "Wireless_Thermometer.amx" an, falls Sie die schrittweise Erstellung des Applikations-Scripts in Phase 2 überspringen wollen. Diese ist im Paket der ergänzenden Dateien zu diesem Tutorial enthalten.

5 Eingabefeld zur Konfiguration des Connectors, der eine Nutzung der durch das Script erzeugten Daten und bereitgestellten Konfigurationen in Verbindung mit der Oberfläche des Cloud-Servers ermöglicht

Kopieren Sie in dieses Feld nacheinander die in den folgenden 3 Schritten beschriebenen Blöcke. Wollen Sie diese schrittweise Erstellung des Connectors überspringen, können Sie auch den gesamten Inhalt der Datei "Datenstruktur.txt" in dieses Feld kopieren und das Tutorial bei Schritt 11 fortsetzen. Die Datei "Datenstruktur.txt" ist im Paket der ergänzenden Dateien zu diesem Tutorial enthalten.

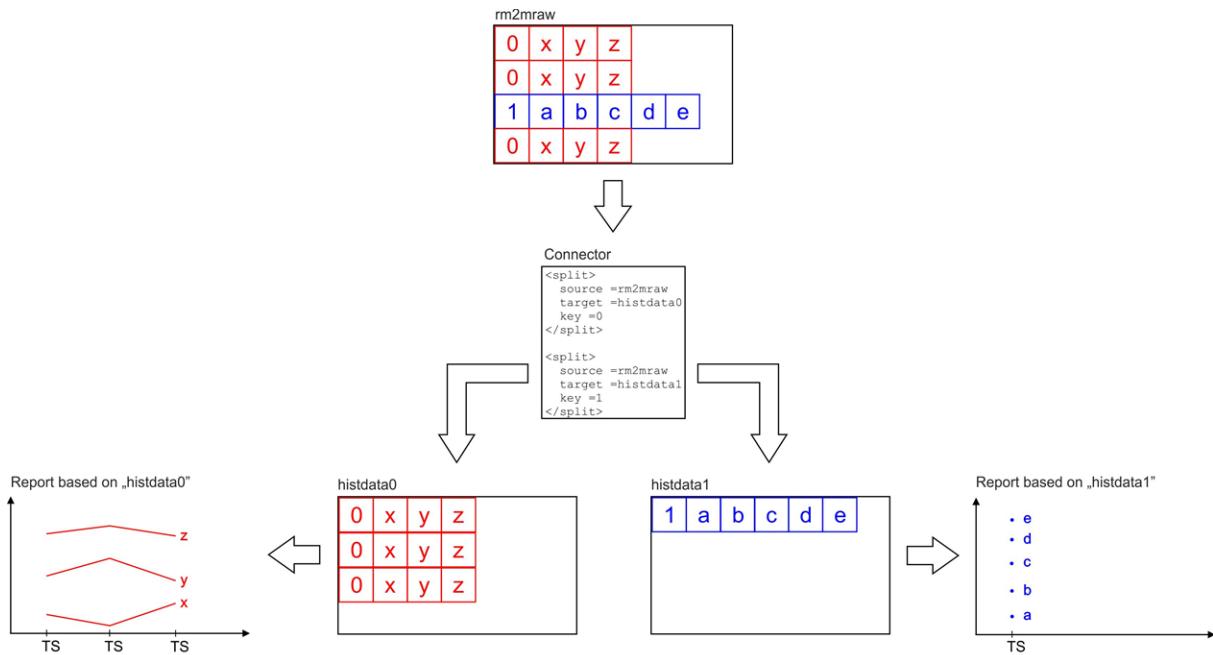
8. Zunächst gilt es zu überlegen welche Konfigurationsparameter für die gewünschte Funktion erforderlich sind. Im konkreten Fall sind dies ein Parameter zur Auswahl des Übertragungsintervalls, einer zur Auswahl des Aufzeichnungsintervalls und einer zur Auswahl der Verbindungsart (Intervall, Wakeup oder Online). Die folgende Tabelle enthält zusätzliche Anforderungen für die Definition der Struktur der Konfigurationsparameter:

Parameter	Datentyp	Einheit/Auswahlmöglichkeiten	Kommastellen	Defaultwert
Übertragungsintervall	32Bit unsigned	min.	keine	60min.
Aufzeichnungsintervall	32Bit unsigned	min.	keine	1min.
Verbindungsart	8Bit unsigned	0 = Intervall 1 = Wakeup 2 = Online	keine	Intervall

Aufgrund der Entscheidung den Konfigurationsblock 0 zu verwenden ergibt sich folgende Datenstruktur für die Konfiguration:

```
<table>
    //Der Konfigurationsblock 0 soll verwendet werden
    name = config
    //Bezeichnung für den Konfig. Abschnitt, die am Server angezeigt wird
    title = Configuration
    <field>
        //Parameter 0 soll für das Übertragungsintervall verwendet werden
        name      = field0
        //Alternativer Feldname, der von der REST-API verwendet werden kann
        alias     = RecordInterval
        //Bezeichnung des Parameters, die am Server angezeigt wird
        title     = Record Interval
        //Position im Konfigblock 0 an der der Parameter gespeichert ist
        byteofs   = 0
        //keine Kommastellen
        decpl     = 0
        //Datentyp: 32Bit unsigned
        type      = u32
        //Einheit in der der Wert für den Parameter einzugeben ist
        units     = sec
        //kleinster gültiger Wert für den Parameter
        min       = 10
        //Defaultwert ist 60sec.
        default   = 60
    </field>
    <field>
        //Parameter 1 soll für das Aufzeichnungsintervall verwendet werden
        name      = field1
        alias     = TransmissionInterval
        title     = Transmission Interval
        byteofs   = 4
        decpl     = 0
        type      = u32
        units     = min
        min       = 10
        default   = 60
        //Umrechnungsfaktor sec.->min. (intern wird mit sec. gearbeitet)
        vscale    = 0.0166666667
    </field>
    <field>
        //Parameter 2 soll für die Verbindungsart verwendet werden
        name      = field2
        alias     = TransmissionMode
        title     = Transmission Mode
        byteofs   = 8
        decpl     = 0
        type      = u8
        default   = 0
        //Auswahlmöglichkeiten für die Dropdown-Liste
        editmask  = 0=Intervall;1=Wakeup;2=Online
    </field>
</table>
```

9. Als nächstes muss festgelegt werden, wie viele und welche Messdatenkanäle des Cloud-Servers verwendet werden sollen. Das rapidM2M M3 kann Datensätze mit unterschiedlicher Länge und Struktur in seinem internen Speicher ablegen. Der Cloud-Server hingegen benötigt immer Datensätze mit gleichbleibendem Aufbau. Daher müssen die empfangenen Datensätze gemäß ihres Aufbaus auf die einzelnen Messdatenkanäle des Cloud-Servers verteilt werden. Dies erfolgt mittels des "Split-tags".



Graphische Darstellung der Funktionsweise der Split-tags

Im aktuellen Beispiel werden die Datensätze immer die selbe Länge und Struktur besitzen wodurch nur einer der Messdatenkanäle des Cloud-Servers benötigt wird. In Kombination mit der Entscheidung den Messdatenkanal 0 zu verwenden, ergibt sich folgende Struktur für den Split-tag:

```
<split>
  //Rohdatenkanal (beim rapidM2M M3 immer "rm2mraw") aus dem die Datensätze kopiert werden sollen
  source = rm2mraw
  //Messdatenkanal in den der Datensatz kopiert werden soll
  target = histdata0
  //Entspricht das erste Byte des Datensatzes diesem Wert, wird er in den angegeben Messdatenkanal
  //kopiert.
  key    = 00
</split>
```

10. Zuletzt muss noch definiert werden welche Messwerte durch die zu entwerfende Applikation erfasst werden. In diesem Beispiel sind das die Batteriespannung, die USB-Ladespannung und die Temperatur. Die folgende Tabelle enthält zusätzliche Anforderungen für die Definition der Struktur der Messwerte:

Messwert	Datentyp	Einheit	Kommastellen
Batteriespannung	16Bit signed	V	2
USB-Ladespannung	16Bit signed	V	2
Temperatur	16Bit signed	°C	1

Aufgrund der bereits im vorherigen Schritt getroffenen Entscheidung den Messdatenkanal 0 zu verwenden ergibt sich folgende Datenstruktur für die Messwerte:

```
<table>
    //Der Messdatenkanal 0 soll verwendet werden.
    name = histdata0
    <field>
        //Datenfeld 0 soll für die Batteriespannung verwendet werden.
        name      = ch0
        //Alternativer Feldname, der von der REST-API verwendet werden kann
        alias     = VoltageBattery
        //Bezeichnung des Datenfelds, die am Server angezeigt wird
        title     = Battery Voltage
        //Position im Messdatenkanal 0, an der das Datenfeld gespeichert ist
        //Hinweis: An Position "0" befindet sich der "key" für den Split-Tag
        byteofs   = 1
        //Der Messwert soll auf 2 Kommastellen gerundet werden.
        decpl    = 2
        //Datentyp: 16Bit signed
        type     = s16
        //Umrechnungsfaktor mV->V (intern wird mit V gearbeitet)
        vscale   = 0.001
        //Einheit des Messwertes, die am Server angezeigt wird
        units    = V
    </field>
    <field>
        //Datenfeld 1 soll für die USB-Ladespannung verwendet werden.
        name      = ch1
        alias     = VoltageUSB
        title     = USB Voltage
        byteofs   = 3
        decpl    = 2
        type     = s16
        vscale   = 0.001
        units    = V
    </field>
    <field>
        //Datenfeld 2 soll für die Temperatur verwendet werden.
        name      = ch2
        alias     = Temperature
        title     = Temperature
        byteofs   = 5
        decpl    = 1
        type     = s16
        vscale   = 0.1
        units    = °C
    </field>
</table>
```

11. Verlassen Sie die Eingabemaske zur Konfiguration der Messstelle durch Klicken des Buttons "speichern". Nach dem Speichervorgang können die durch den Connector spezifizierten Konfigurationsparameter und Messkanäle in der Oberfläche des Cloud-Servers verwendet werden.
12. Öffnen Sie die Eingabemaske zur Konfiguration der Messstelle erneut durch Klicken auf den Namen der Messstelle in der Liste der Messstellen / Applikationen.

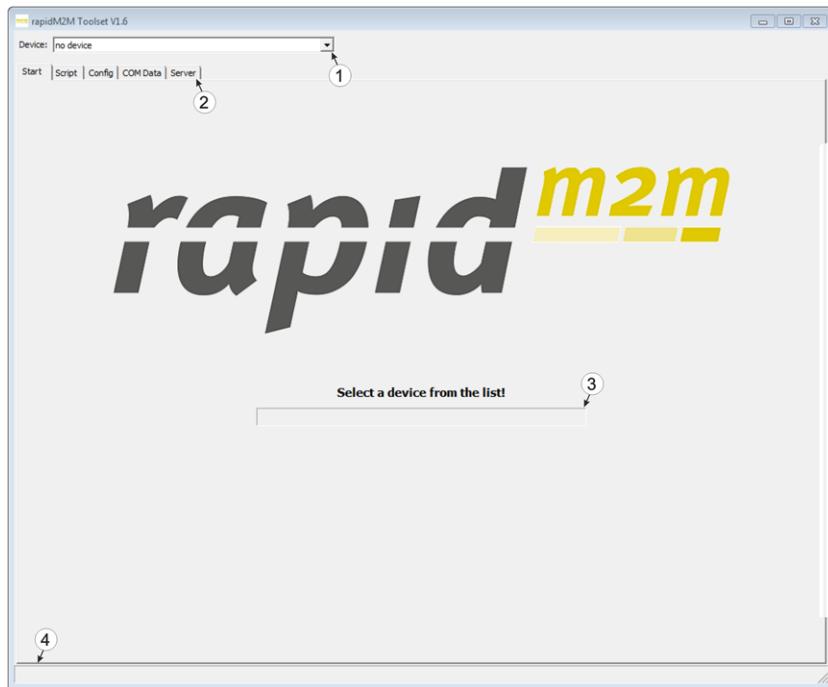
13. Klicken Sie in der Eingabemaske zur Konfiguration der Messstelle auf "Configuration", um die Eingabefelder für das Aufzeichnungsintervall, das Übertragungsintervall und die Verbindungsart einzublenden. Setzen Sie die Parameter wie folgt. Ändern Sie auch gleich den Namen der Messstelle um spätere Verwechslungen zwischen Messtellenname und Seriennummer zu vermeiden. Verlassen Sie anschließend die Eingabemaske durch Klicken des Buttons "speichern".

Eingabemaske zur Konfiguration der Messstelle / Applikation, Abschnitt "Configuration"

1 Name der Messstelle (frei wählbar)
Wählen Sie hier "Wireless Thermometer"
2 Konfigurationsabschnitt "Configuration"
3 zeitlicher Abstand der Messdatenaufzeichnungen
Wählen Sie hier "10sec.".
4 zeitlicher Abstand der Übertragungen (nur relevant für die Verbindungsarten "Intervall" und "Wakeup")
Wählen Sie hier "60min.".
5 Auswahl der Verbindungsart
Wählen Sie hier "online". Das Gerät trennt die Verbindung nicht und übermittelt kontinuierlich die Messdaten.

2.2 Phase 2: Erstellen des Applikations-Scripts

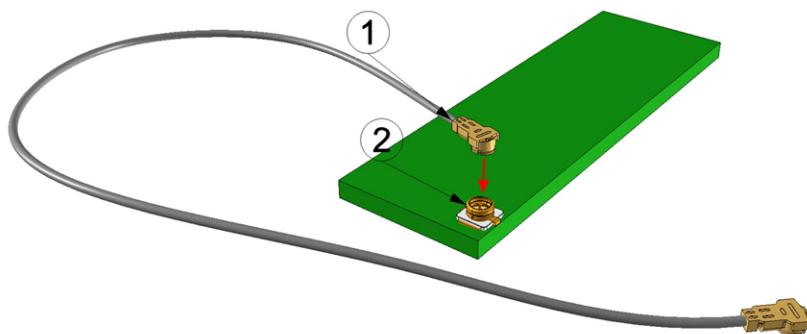
1. Installieren Sie das rapidM2M Toolset . Es steht unter folgender Adresse gratis zum Download bereit: www.microtronics.at/Toolset.
2. Starten Sie das rapidM2M Toolset .



Karteireiter "Start" (kein rapidM2M M3 ausgewählt)

1 Auswahl des rapidM2M M3	3 Fortschrittsbalken für den Aufbau der USB-Verbindung zum rapidM2M M3
2 Auswahl der Karteireiter für die verschiedenen Funktionen des rapidM2M Toolset , die auch ohne verbundenem rapidM2M M3 verfügbar sind	4 Anzeige des Kommunikationsstatus auf der USB-Schnittstelle zum rapidM2M M3

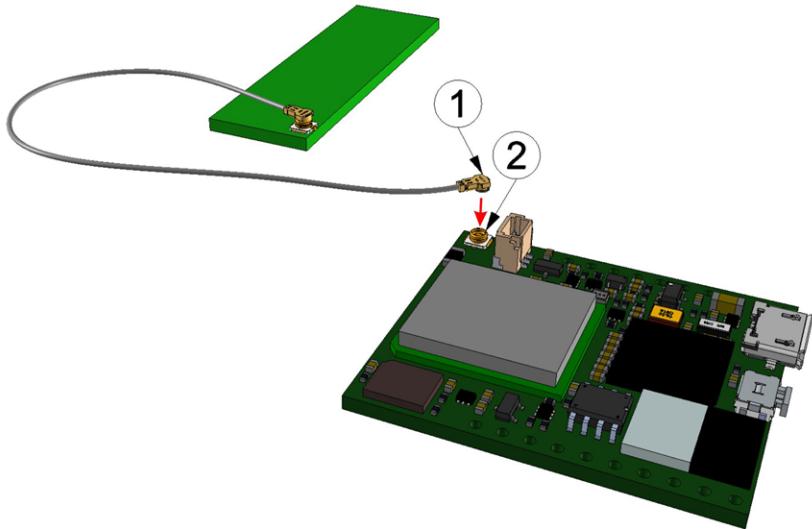
3. Verbinden Sie das U.FL Antennenkabel mit dem U.FL Antennenanschluss der PCB Antenne.



Antennenkabel mit Antenne verbinden

1 U.FL Antennenkabel	2 U.FL Antennenanschluss der PCB Antenne
-----------------------------	---

-
4. Verbinden Sie das U.FL Antennenkabel mit dem U.FL Antennenanschluss des rapidM2M M3

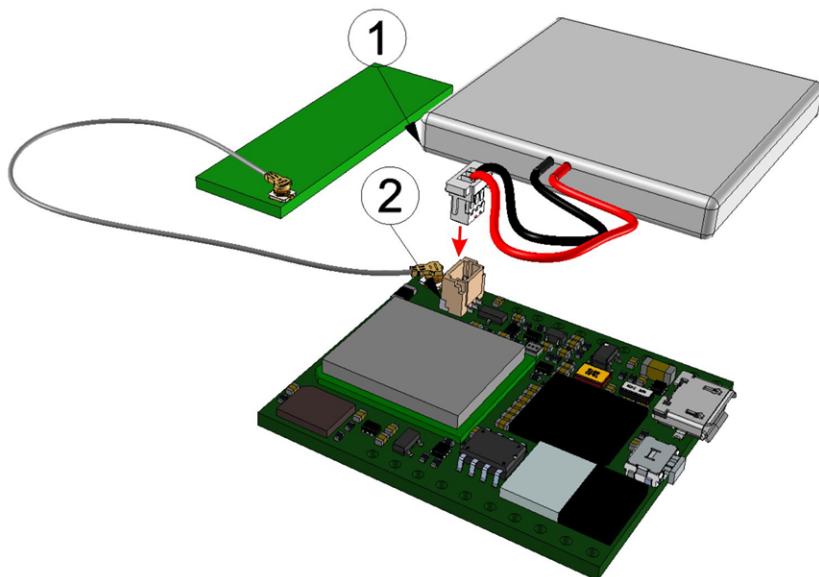


Antennenkabel mit dem rapidM2M M3 verbinden

1 U.FL Antennenkabel	2 U.FL Antennenanschluss des rapidM2M M3
----------------------	--

5. Verbinden Sie den Akku mit dem Akku-Anschluss des rapidM2M M3 . Wurde noch kein Script in das rapidM2M M3 geladen sollte daraufhin die LED grün zu flackern beginnen, um den Verbindungsaufbau zu signalisieren.

Hinweis: Sorgen Sie vor dem Anschluss des Akkus dafür, dass sich keine Gegenstände aus blankem Metall in der Nähe des rapidM2M M3 befinden, da diese zu einem Kurzschluss führen könnten.

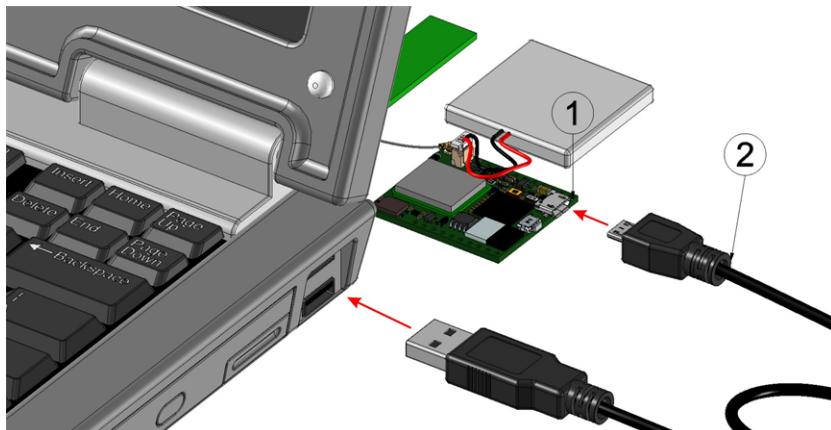


Akku mit dem rapidM2M M3 verbinden

1 Li-Po Akku	2 Akku-Anschluss des rapidM2M M3
--------------	----------------------------------

- Verbinden Sie das rapidM2M M3 mit Hilfe eines USB-Kabels (A-Stecker auf Micro-B-Stecker) mit Ihrem PC und warten Sie bis die automatische Installation der für das rapidM2M M3 benötigten Treiber abgeschlossen ist. Über die USB-Schnittstelle wird auch der Li-Po Akku des rapidM2M M3 geladen.

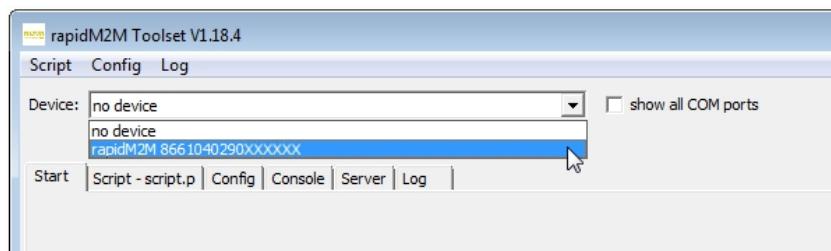
Hinweis: Bei der ersten Inbetriebnahme sollte der Li-Po Akku ca. 12h geladen werden.



rapidM2M M3 mit dem PC verbinden

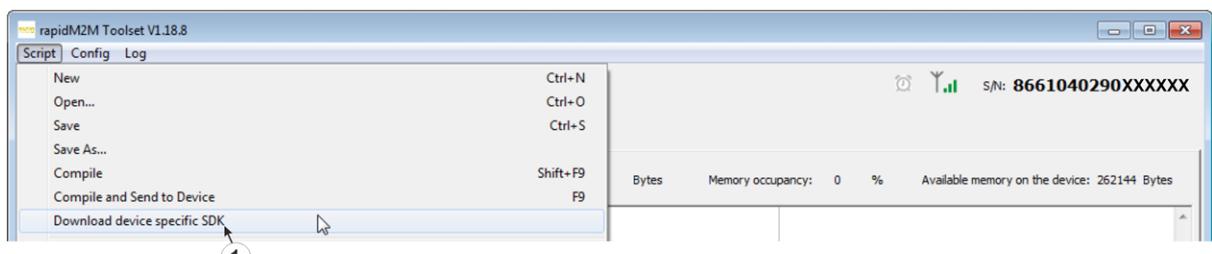
1 Micro-USB Buchse des rapidM2M M3	2 USB-Kabel (A-Stecker auf Micro-B-Stecker)
------------------------------------	---

- Wechseln Sie zum rapidM2M Toolset und wählen Sie Ihr rapidM2M M3 anhand der Seriennummer aus der Liste der gefundenen Geräte aus.



Geräteauswahl

- Laden Sie die spezifischen Erweiterungen für das rapidM2M M3 über den Menüpunkt "Script → Download device specific SDK" herunter.

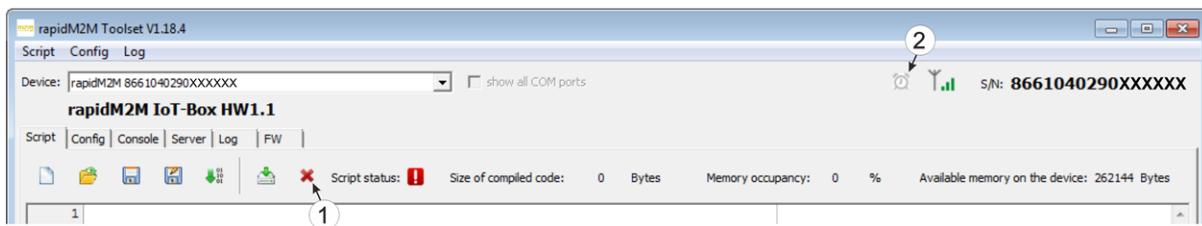


Gerätespezifisches SDK herunterladen

1 Herunterladen bzw. Aktualisieren der spezifischen Erweiterungen für das aktuell mit dem PC verbundene Gerät

9. Löschen Sie das im rapidM2M M3 installierte Script (-Symbol). Die Anzeige des GSM-Verbindungsstatus wechselt nach kurzem Blinken (Verbindung wird hergestellt) zunächst auf grün (verbunden) und schließlich auf das -Symbol (offline, Wakeup über den Cloud-Server möglich).

Hinweis: Wenn Sie in Phase 1 bereits das Script Binary "Wireless_Thermometer.amx" auf den Cloud-Server geladen haben, d.h. sich für das Überspringen der schrittweisen Erstellung des Applikations-Scripts entschieden haben, wechselt der GSM-Verbindungsstatus nach grün (verbunden) auf grau (offline). Setzen Sie in diesem Fall das Tutorial bei Schritt 22 fort.



Löschen des im rapidM2M M3 installierten Scripts

- | | |
|---|--|
| 1 | löscht das installierte Script aus dem rapidM2M M3 |
| 2 | Anzeige des GSM-Verbindungsstatus. Durch Klicken auf dieses Symbol kann ein Verbindungsauftakt gestartet werden. |

GSM-Verbindungsstatus

Statussymbol	
	offline
	offline, Wakeup über den Cloud-Server möglich
	Verbindung wird hergestellt
	verbunden
	letzter Verbindungsversuch ist fehlgeschlagen
	letzter Verbindungsversuch ist fehlgeschlagen, Wakeup über den Cloud-Server möglich

10. Wählen Sie nun den Karteireiter "Config" und löschen Sie auch die Inhalte der Konfigurationsblöcke des rapidM2M M3 (-Symbol).



Löschen der Inhalte der Konfigurationsblöcke

- | | |
|---|---|
| 1 | löscht die Inhalte aller Konfigurationsblöcke des rapidM2M M3 |
|---|---|

11. Wechseln Sie zurück zum Karteireiter "Script", erstellen Sie ein neues leeres Script (-Symbol) und speichern Sie es auf Ihrer Festplatte (-Symbol).



Erstellen eines neuen Scripts

- 1 Dateiname des geöffneten Scripts. Wurde das Script geändert jedoch noch nicht gespeichert, wird dies durch "*" nach dem Dateinamen signalisiert.
- 2 erstellt ein neues leeres Script
- 3 öffnet den Dialog zum Speichern des Scripts unter einem anderen Dateinamen
- 4 Script-Editor

Kopieren Sie in dieses Feld nacheinander die in den folgenden 9 Schritten beschriebenen Blöcke. Wollen Sie diese schrittweise Erstellung des Applikations-Scripts überspringen, können Sie auch den gesamten Inhalt der Datei "Wireless_Thermometer.p" in dieses Feld kopieren und das Tutorial bei Schritt 21 fortsetzen. Die Datei "Wireless_Thermometer.p" ist im Paket der ergänzenden Dateien zu diesem Tutorial enthalten.

12. Zunächst gilt es den Pfad für das spezifische Include-File für das rapidM2M M3 und die Funktionsprototypen der von der Firmware bei entsprechenden Events aufgerufenen Callbacks anzugeben.

```
/* Pfad für das hardwarespezifische Include-File */
#include ".\rapidM2M_IoT-Box\iotbox"

/* Forward Deklarationen der öffentlichen Funktionen */
forward public Timerlsl();                                // für den Programmablauf, wird 1x pro sec. aufgerufen
forward public ReadConfig(cfg);                            // wird bei Änderung eines der Konfigblöcke aufgerufen
forward public KeyChanged(iKeyCode);                      // wird beim Drücken und Loslassen der Taste aufgerufen
```

13. Als nächstes sind einige Konstanten und globale Variablen für den Programmablauf zu definieren.

```
/* Standardwerte für die Initialisierung der Konfiguration */
const
{
    ITV_RECORD      = 1 * 60,           // Aufzeichnungsintervall [sec.], default 1 min
    ITV_TRANSMISSION = 1 * 60 * 60,     // Übertragungsintervall [sec.], default 60 min
    TXMODE          = RM2M_TXMODE_TRIG, // Verbindungsart, default "Intervall"

    DEFAULT_COLOR   = 0x000008ECF,       // Farbe der LED
}

/* Größen- und Indexangaben für die Konfigurationsblöcke und den Messdatenblock */
const
{
    CFG_BASIC_INDEX = 0,               // In Konfigblock 0 befindet sich die Basiskonfig
    CFG_BASIC_SIZE  = 9,               // // Aufzeichnungsintv.(u32) + Übertragungsintv.(u32) +
                                    // // Verbindungsart (u8)

    HISTDATA_SIZE   = 3 * 2 + 1,       // 3 Kanäle (s16) + "Split-tag" (u8)
}

/* Globale Variablen zur Aufnahme der aktuellen Konfiguration */
static iRecItv;                     // aktuelles Aufzeichnungsintervall [sec.]
static iTxItv;                      // aktuelles Übertragungsintervall [sec.]
static iTxMode;                     // aktuelle Verbindungsart (0 = Intervall,
                                    // // 1 = Wakeup, 2 = Online)

/* Globale Variablen für die verbleibende Zeit bis zum Auslösen bestimmter Aktionen */
static iRectimer;                  // sec. bis zur nächsten Aufzeichnung
static iTxTimer;                   // sec. bis zur nächsten Übertragung
```

14. Nun folgt die Erstellung der main-Funktion.

```

main()
{
    /* Zwischenspeicher für den Index einer öffentlichen Funktion und den Rückgabewert einer Funktion */
    new iIdx, iResult;

    /* Initialisierung des Tasters -> Auswertung durch das Script aktiviert
     * - Index der Funktion, die beim Drücken und Loslassen der Taste aufgerufen werden soll, ermitteln
     * - Index ans System übergeben sowie mitteilen, dass die Steuerung der Taste durch das Script erfolgt
     * - Index sowie Rückgabewert der Init-Funktion werden über die Console ausgegeben */
    iIdx = funcidx("KeyChanged");
    iResult = Switch_Init(SWITCH_MODE_SCRIPT, iIdx);
    printf("Switch_Init(%d) = %d\r\n", iIdx, iResult);

    /* Initialisierung der LED -> Ansteuerung durch das Script aktiviert */
    Led_Init(LED_MODE_SCRIPT);

    /* Initialisierung eines lsec. Timers, der für den generellen Programmablauf verwendet wird
     * - Index der Funktion, die 1x pro sec. ausgeführt werden soll, ermitteln
     * - Index ans System übergeben
     * - Index sowie Rückgabewert der Funktion zum Erzeugen des Timers werden über die Console ausgegeben */
    iIdx = funcidx("Timer1s");
    iResult = rM2M_TimerAdd(iIdx);
    printf("rM2M_TimerAdd(%d) = %d\r\n", iIdx, iResult);

    /* Festlegen der Funktion, die bei einer Änderung eines Konfigblockes aufgerufen werden soll
     * - Index der Funktion, bei Änderung eines der Konfigblöcke aufgerufen, ermitteln
     * - Index ans System übergeben
     * - Index sowie Rückgabewert der Init-Funktion werden über die Console ausgegeben */
    iIdx = funcidx("ReadConfig");
    iResult = rM2M_CfgOnChg(iIdx);
    printf("rM2M_CfgOnChg(%d) = %d\r\n", iIdx, iResult);

    /* Auslesen der Basiskonfiguration. Ist der Konfigblock 0 noch leer (erster Programmstart), wird die
     * Basiskonfiguration mit Standardwerten initialisiert. */
    ReadConfig(CFG_BASIC_INDEX);

    /* Durch das Setzen der Zähler auf 0 wird sofort eine Übertragung sowie eine Aufzeichnung ausgelöst.
     * Das 0 setzen könnte auch unterlassen werden, da in PAWN alle Variablen beim Anlegen mit 0
     * initialisiert werden. Es wurde an dieser Stelle zum besseren Verständnis jedoch durchgeführt. */
    iTxTimer = 0;
    iRecTimer = 0;

    /* Setzen der Verbindungsart */
    rM2M_TxSetMode(iTxMode);
}

```

15. Im Anschluss gilt es die Funktion, die 1x pro Sekunde durch das System aufgerufen wird und somit maßgebend für den Programmablauf ist, zu erstellen.

```

/* lsec. Timer, wird für den generellen Programmablauf verwendet */
public Timer1s()
{
    Handle_Led();                                // Steuerung der LED
    Handle_Transmission();                         // Steuerung der Übertragung
    Handle_Record();                             // Steuerung der Aufzeichnung
}

```

16. Jetzt muss die Funktion, die die Ansteuerung der LED übernimmt, erzeugt werden.

```
/* Funktion für die Steuerung der LED */
Handle_Led()
{
    new iTxStatus;                                // Zwischenspeicher für den Verbindungsstatus

    iTxStatus = rM2M_TxGetStatus();                // Aktuellen Verbindungsstatus vom System lesen

    if(iTxStatus & RM2M_TX_ACTIVE)               // Wenn gerade eine GPRS-Verbindung besteht ->
    {
        Led_Off();                                // LED ausschalten
        Led_On(DEFAULT_COLOR);                   // LED einschalten (Default LED-Farbe)
    }

    /* Wenn gerade ein Verbindungsaufbau durchgeführt wird oder die Wartezeit bis zum Retry läuft -> */
    else if(iTxStatus & (RM2M_TX_STARTED|RM2M_TX_RETRY))
    {
        Led_Off();                                // LED ausschalten
        Led_Flicker(0, DEFAULT_COLOR);            // LED flackert dauerhaft (Default LED-Farbe)
    }

    else if( iTxStatus & RM2M_TX_FAILED)          // Wenn der letzte Verbindungsaufbau fehlgeschlagen ist ->
    {
        Led_Off();                                // LED ausschalten
        Led_Blink(0, 0x00FF0000);                 // LED blinkt dauerhaft rot
    }

    else
        Led_Off();                                // Andernfalls ->
                                                // LED ausschalten
}
}
```

17. Die nächste Funktion, die generiert werden muss, übernimmt die Behandlung des Übertragungsintervalls.

```
/* Funktion für die Erzeugung des Übertragungsintervalls */
Handle_Transmission()
{
    iTxTimer--;                                // Zähler mit den sec. bis zur nächsten Übertragung reduzieren
    if(iTxTimer <= 0)                          // Wenn der Zähler abgelaufen ist ->
    {
        rM2M_TxStart();                        // Eine Verbindung zum Server herstellen
        iTxTimer = iTxItv;                      // Zählervar. auf akt. Übertragungsintv. [sec.] zurücksetzen
    }
}
}
```

18. Danach ist die Funktion zu entwerfen, die die Aufzeichnung der Daten übernimmt.

```
/* Funktion für die Aufzeichnung der Daten */
Handle_Record()
{
    /* Zwischenspeicher in dem der zu speichernde Datensatz zusammengesetzt wird. */
    new aRecData{HISTDATA_SIZE};

    iRecTimer--;                                // Zähler mit sec. bis zur nächsten Aufzeichnung reduzieren
    if(iRecTimer <= 0)                          // Wenn der Zähler abgelaufen ist ->
    {
        new aSysValues[TIoTbox_SysValue];        // Zwischenspeicher für die int. Messwerte (VBat, VUsb, Temp)

        IoTbox_GetSysValues(aSysValues);          // Auslesen der aktuellen int. Messwerte (VBat, VUsb, Temp)

        /* Zusammensetzen des zu speichernden Datensatzes im Zwischenspeicher "aRecData"
         * Das erste Byte(Position 0 im Array "aRecData") wird auf 0 gesetzt damit der Server den Datensatz,
         * wie beim Entwurf des Connectors angegeben, beim Empfangen in Messdatenkanal 0 kopiert
         * Auf Postion 1-2 wird die Batteriespannung (VBat) kopiert. Datentyp: s16
         * Auf Postion 3-4 wird die USB-Ladespannung (VUsb) kopiert. Datentyp: s16
         * Auf Postion 5-6 wird die Temperatur (Temp) kopiert. Datentyp: s16
         */
        aRecData{0} = 0;                         // "Split-tag"
        rM2M_Pack(aRecData, 1, aSysValues.VBat, RM2M_PACK_BE + RM2M_PACK_S16);
        rM2M_Pack(aRecData, 3, aSysValues.VUsb, RM2M_PACK_BE + RM2M_PACK_S16);
        rM2M_Pack(aRecData, 5, aSysValues.Temp, RM2M_PACK_BE + RM2M_PACK_S16);

        /* Zusammengesetzten Datensatz ans System zur Aufzeichnung übergeben */
        rM2M_RecData(0, aRecData, HISTDATA_SIZE);

        iRecTimer = iRecItv;                      // Zählervariable auf akt. Aufzeichnungsintv. zurücksetzen
    }

    /* Aktuelle Messwerte über die Console ausgegeben */
    printf("Vb:%d Vu:%d Ti:%d\r\n", aSysValues.VBat, aSysValues.VUsb, aSysValues.Temp);
}
}
```

-
19. Anschließend erfolgt die Erstellung der Funktion, die beim Drücken und Loslassen der Taste durch das System aufgerufen wird und mittels jener das Auslösen einer Übertragung durch einen Tastendruck ermöglicht wird.

```
/* Funktion, die das Auslösen einer Übertragung durch einen Tastendruck ermöglicht */
public KeyChanged(iKeyState)
{
    printf("K:%d\r\n", iKeyState); // Aktion über die Console ausgeben (0=Loslassen, 1=Drücken)

    if(!iKeyState) // Wenn die Taste losgelassen wurde ->
    {
        /* Zählvariable der sec. bis zur nächsten Übertragung auf 0 setzen. Dadurch wird beim nächsten Aufruf
           der Funktion "Handle_Transmission" durch die Funktion "Timer1s" eine Übertragung ausgelöst. */
        iTxTimer = 0;
    }
}
```

20. Zuletzt ist es noch erforderlich jene Funktion zu erzeugen, die bei jeder Änderung eines der Konfigblöcke durch das System aufgerufen wird und es somit ermöglicht auf eine vom Server empfangene, geänderte Konfiguration zu reagieren.

```

/* Funktion, die es ermöglicht auf eine vom Server empfangene, geänderte Konfiguration zu reagieren */
public ReadConfig(cfg)
{
    // Wenn es sich bei der geänderten Konfiguration um die Basiskonfig handelt -> *
    if(cfg == CFG_BASIC_INDEX)
    {
        new aData[CFG_BASIC_SIZE];           // Zwischenspeicher für die vom System gelesene Basiskonfig
        new isize;                          // Zwischenspeicher für die Größe der Basiskonfig in Bytes
        new iTmp;                           // Zwischenspeicher für einen Parameter der Basiskonfig

        /* Basiskonfig vom System lesen und in den Zwischenspeicher kopieren. Anschließend wird die
         * Nummer des Konfigblocks und der Rückgabewert der Lesefunktion (Anzahl der Bytes bzw. Fehlercode)
         * über die Console ausgegeben
        */
        isize = rM2M_CfgRead(cfg, 0, aData, CFG_BASIC_SIZE);
        printf("Cfg %d size = %d\r\n", cfg, isize);

        /* Wenn die Anzahl der gelesenen Bytes unter der Größe der Basiskonfig liegt ->
         * Info: Fehlercodes sind negativ
        */
        if(isize < CFG_BASIC_SIZE)
        {
            /* Der folgende Block kopiert zunächst die Default-Werte in den Zw.speicher für die gelesene
             * Basiskonfig und setzt dann den Zw.speicher für die Größe der Basiskonfig auf die aktuelle
             * Größe. Dadurch wird das folgende "IF" sowohl beim Neuinitialisieren als auch bei empfangenen
             * Änderungen durchlaufen. Müssen für einzelne Parameter spezielle Aktionen ausgeführt werden,
             * ist es so nicht notwendig dies für beide Fälle getrennt zu implementieren.
            */
            iTmp = ITV_RECORD;
            rM2M_Pack(aData, 0, iTmp, RM2M_PACK_BE + RM2M_PACK_U32);
            iTmp = ITV_TRANSMISSION;
            rM2M_Pack(aData, 4, iTmp, RM2M_PACK_BE + RM2M_PACK_U32);
            iTmp = TXMODE;
            rM2M_Pack(aData, 8, iTmp, RM2M_PACK_BE + RM2M_PACK_U8);
            isize = CFG_BASIC_SIZE;
            printf("created new Config #0\r\n");
        }
    }

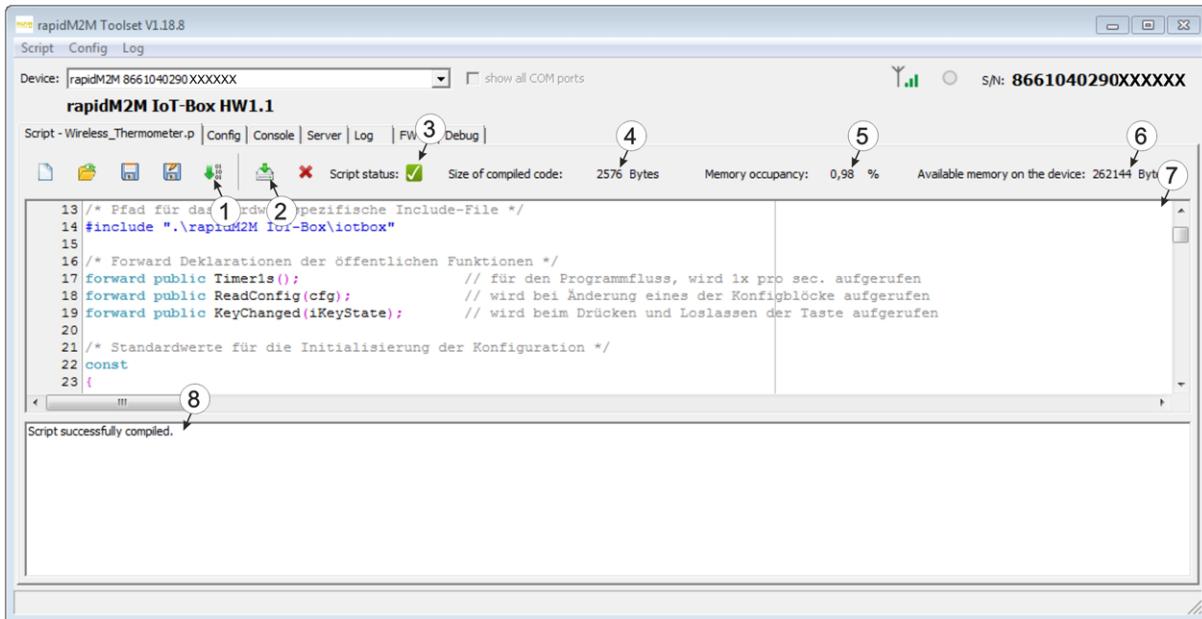
    /* Wenn die Anzahl der gelesenen Bytes mindestens der Größe der Basiskonfig entspricht -> */
    if(isize >= CFG_BASIC_SIZE)
    {
        /* Aufzeichnungsintervall(u32, Big Endian) auf Postion 0-3 aus dem Zw.speicher für die gelesene
         * Basiskonfig in den Zw.speicher für einen Parameter kopieren
        */
        rM2M_Pack(aData, 0, iTmp, RM2M_PACK_BE + RM2M_PACK_U32 + RM2M_PACK_GET);
        if(iTmp != iRecItv)      // Wenn der empfangene Wert nicht jenem der globalen Variablen entspricht ->
        {
            printf("iRecItv changed to %d s\r\n", iTmp); // empfangenen Wert über die Console ausgeben
            iRecItv = iTmp;                                // empfangenen Wert in globale Variable übernehmen
        }

        /* Übertragungsintervall(u32, Big Endian) auf Postion 4-7 aus dem Zw.speicher für die gelesene
         * Basiskonfig in den Zw.speicher für einen Parameter kopieren
        */
        rM2M_Pack(aData, 4, iTmp, RM2M_PACK_BE + RM2M_PACK_U32 + RM2M_PACK_GET);
        if(iTmp != iTxItv)      // Wenn der empfangene Wert nicht jenem der globalen Variablen entspricht ->
        {
            printf("iTxAItv changed to %d s\r\n", iTmp); // empfangenen Wert über die Console ausgeben
            iTxItv = iTmp;                                // empfangenen Wert in globale Variable übernehmen
        }

        /* Verbindungsart(u8) auf Postion 8 aus dem Zw.speicher für die gelesene Basiskonfig in den
         * Zw.speicher für einen Parameter kopieren
        */
        rM2M_Pack(aData, 8, iTmp, RM2M_PACK_BE + RM2M_PACK_U8 + RM2M_PACK_GET);
        if(iTmp != iTxMode)      // Wenn der empfangene Wert nicht jenem der globalen Variablen entspricht ->
        {
            rM2M_TxSetMode(iTmp);                      // Verbindungsart auf den empfangenen Wert setzen
            printf("iTxAItv changed to %d s\r\n", iTmp); // empfangenen Wert über die Console ausgeben
            iTxMode = iTmp;                            // empfangenen Wert in globale Variable übernehmen
        }
    }
}

```

21. Kompilieren und Laden Sie das soeben erstellte Script mittels des -Symbols in das rapidM2M M3 . Die Status-LED des rapidM2M M3 beginnt nach dem Script-Download blau zu flackern, um den Verbindungsaufbau zu signalisieren.



Karteireiter "Script"

- | | |
|----------|--|
| 1 | kompiliert das geöffnete Script. Dabei wird es auch automatisch gespeichert. |
| 2 | überträgt das geöffnete Script in das Modul/Gerät. Dabei wird das Script zuerst gespeichert und dann kompiliert. Das kompilierte PAWN-Binary (*.amx) wird im selben Ordner wie das Script gespeichert. |
| 3 | Status der Script-Übertragung |
| 4 | Größe des kompilierten Codes |
| 5 | gibt die Belegung des verfügbaren Speichers in % an |
| 6 | gibt die Größe des verfügbaren Speichers am rapidM2M M3 an (Angabe der komprimierten Größe) |
| 7 | Script-Editor |
| 8 | Debugausgabe des Script-Kompilers. Durch einen Doppelklick auf einen Eintrag in der Debugausgabe springt der Cursor im Script-Editor zur betroffenen Zeile. |

Script Status

Symbol	Beschreibung
	Script wurde erfolgreich in das rapidM2M M3 geladen.
	Script wird gerade zum rapidM2M M3 übertragen.
	Der Zeitstempel des im rapidM2M M3 gespeicherten kompilierten Scripts stimmt nicht mit dem des kompilierten Scripts des rapidM2M Toolset überein.

22. Wechseln Sie zum Cloud-Server und warten Sie bis in der Liste der Messstellen / Applikationen angezeigt wird, dass das rapidM2M M3 mit dem Cloud-Server verbunden ist (rotierende Pfeile). Dies kann bis zu 2 Minuten dauern.



Liste der Messstellen / Applikationen

- 1 Zeigt an, dass das rapidM2M M3 mit dem Server verbunden ist und Daten überträgt. Dieses Symbol wird ausgeblendet, wenn die Verbindung getrennt wird.
- 2 Informationen zum Zeitpunkt der Kommunikation zwischen Messgerät und Server
 - Messgerät ist mit dem Server verbunden und überträgt Daten: Zeitpunkt des letzten Verbindungsaufbaus
 - Messgerät ist momentan nicht mit dem Server verbunden: Zeitpunkt des letzten Verbindungsabbaus
- 3 Das Symbol gibt die Signalstärke an. Ein Klick auf das Symbol zeigt Informationen zur Funkzelle, die bei der letzten Verbindung verwendet wurde sowie zu den bei der letzten Positionsbestimmung berücksichtigten Funkzellen an.

23. Die Messwerte werden nun bereits erfasst und aufgrund der Verbindungsart „online“ sofort nach der Ermittlung zum Cloud-Server übertragen. Sobald der erste Datensatz eingetroffen ist, wird in der Anzeige des zur Verfügung stehenden Datenzeitraums anstelle von "KEINE DATEN" der Zeitraum angegeben, für den Daten zur Verfügung stehen.



Liste der Messstellen / Applikationen

- 1 Anzeige des zur Verfügung stehenden Datenzeitraums

2.3 Phase 3: Visualisieren der Daten am Cloud-Server

1. Legen Sie zur Darstellung der Messwerte eine Auswertung an.

The screenshot shows the Microtronics cloud server interface. At the top, there is a navigation bar with links for 'support@microtronics.at', 'abmelden support@microtronics.at', 'API Hilfe', and search functions. Below the navigation bar, there is a sub-navigation menu with 'Messtellen / Applikationen' (highlighted in yellow), 'Auswertungen' (circled in red with number 1), 'Geräte', 'Benutzer', 'Alarme', 'Statistik', and 'Meine Daten'. A message 'Seiten: 1 (Gesamt 0)' and '(keine Einträge)' is displayed. The main content area has tabs for 'Konten', 'Konto-Art', 'Kontostand', and 'Kontostand inkl. 20% MwSt'. Below this, a specific entry for 'support@microtronics.at' is shown with details: 'Vorauszahlung', a green plus sign icon, '5,65 €', and '6,78 €'. Another tab 'Messstellen / Applikationen' is visible, showing a list with 'Wireless Thermometer' (rapidM2M: 8661040290XXXXXX) and a timestamp '12.5.2017 13:45:25 SER UTC+2,00'. On the right side, there are icons for 'Verbindung | App.', file management, signal strength, and search.

Bereich "Messstellen / Applikationen", in dem alle zur Verfügung stehenden Messstellen angezeigt werden

1 öffnet die Eingabemaske zum Erstellen einer neuen Auswertung	2 Liste der Auswertungen
--	--------------------------

Bei "Auswertung erstellen" wählen Sie "Leere Auswertung".

The screenshot shows a modal dialog box titled 'Auswertung erstellen' (Create Measurement). It contains a dropdown menu labeled 'Leere Auswertung' (empty measurement) with a circled '1'. Below the dropdown are two buttons: 'abbrechen' (cancel) and 'erstellen' (create). A large yellow callout points to the 'erstellen' button with a circled '2'. In the background, the main interface is visible with a yellow box highlighting the 'Auswertungen' section. A third yellow callout points to an input field with a circled '3'.

Auswertungsvorlage wählen

1 Dropdown-Liste der verfügbaren Auswertungsvorlagen	3 Eingabefenster zur Auswahl der Auswertungsvorlage
2 Button "erstellen"	

2. Konfigurieren Sie die Anzeigeelemente "letzte Messwerte" wie folgt. Dadurch werden die letzten Messwerte der Messkanäle "Temperature", "Batery Voltage" und "USB Voltage" in der Liste der Auswertungen neben dem Namen der Auswertung angezeigt. Durch Klicken auf den Button "speichern" gelangen Sie wieder zum Bereich "Messstellen / Applikationen", in dem alle zur Verfügung stehenden Messstellen angezeigt werden.

	Messstelle	Kanal	Typ	Min	Max
1	Wireless Thermometer	Temperature	Vertikale Skala	0	45
2	Wireless Thermometer	Batery Voltage	Runde Skala	0	5
3	Wireless Thermometer	USB Voltage	Ziffern		

Eingabemaske zur Grundkonfiguration der Auswertung

- | | |
|---|---|
| 1 | frei wählbarer Name für die Auswertung |
| 2 | Konfiguration der Anzeigeelemente "letzte Messwerte", die in der Liste der Auswertungen neben dem Namen der Auswertung angezeigt werden |
| 3 | Auswahl der Messstelle. Das Element kann Daten von unterschiedlichen Messstellen beziehen. Es stehen alle Messstellen des aktuellen Kunden zur Verfügung. |
| 4 | Kanal der Messstelle, der angezeigt werden soll |
| 5 | Auswahl des Typs des zu verwendenden Anzeigeelements |
| 6 | unteres Skalenende des Anzeigeelements (nur für Skala und Balken-Elemente) |
| 7 | oberes Skalenende des Anzeigeelements (nur für Skala und Balken-Elemente) |

3. Direkt neben dem Namen der Auswertung wird nun der aktuelle Messwert angezeigt. Die Messwerte werden im Aufzeichnungsintervall (10sec. wenn Sie den in diesem Tutorial empfohlenen Wert verwenden), welches Sie über den Konfigurationsabschnitt "Configuration" einstellen können, aktualisiert. Durch Änderung der Temperatur am Temperatursensor (z.B. mittels Kältespray) oder Herstellen bzw. Trennen der USB-Verbindung wird eine Veränderung der Messwerte erreicht.

Wichtiger Hinweis: Bitte beachten Sie, dass die empfindlichen elektronischen Komponenten des rapidM2M M3 bei Berührung durch statische Elektrizität beschädigt werden können.

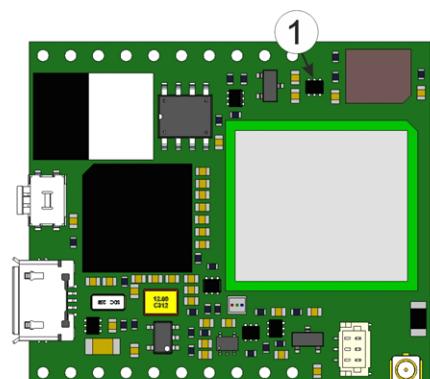
Hinweis: Durch den mit der Verbindungart "Online" verbundenen erhöhten Energieverbrauch und der daraus resultierenden Eigenerwärmung des rapidM2M M3 kann es zu Abweichungen zwischen der angezeigten Temperatur mit der tatsächlichen Umgebungstemperatur kommen.

The screenshot shows the microtronics.at web interface with the following sections:

- Auswertungen:** Shows a summary of measurements. Item 1 points to the 'Auswertung' button. Item 2 points to the 'Auswertung' link. Item 3 points to the measurement details for 'Temperature Wireless Thermome' (Value: 33,7 °C).
- Konten:** Shows account information for 'support@microtronics.at'. Item 1 points to the account name. Item 2 points to the account balance (5,65 €). Item 3 points to the account balance including VAT (6,78 €).
- Messstellen / Applikationen:** Shows a list of measurement stations. Item 1 points to the station name 'Wireless Thermometer'. Item 2 points to the station ID 'rapidM2M: 8661040290XXXXXX'. Item 3 points to the last update timestamp '12.5.2017 15:52:39 SER UTC+2,00'.

Bereich "Messstellen / Applikationen", in dem alle zur Verfügung stehenden Messstellen angezeigt werden

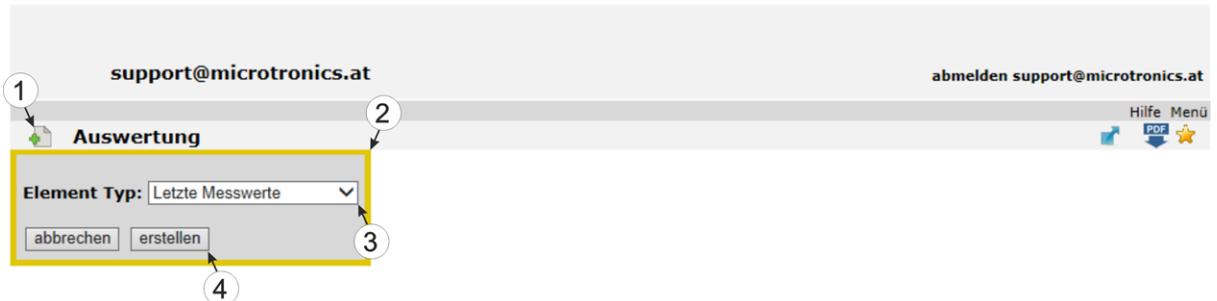
1 öffnet die Eingabemaske zur Grundkonfiguration der Auswertung
2 Durch Klicken auf den Namen der Auswertung wird die Eingabemaske zur Konfiguration und der Auswahl der Elemente geöffnet.
3 Anzeigeelemente für den letzten Messwert eines Messkanals (max. 4 pro Auswertung). Die Auswahl der Messstelle, des Kanals und des Typs des Anzeigeelements erfolgt in der Grundkonfiguration der Auswertung.



rapidM2M M3

- | |
|--------------------|
| 1 Temperatursensor |
|--------------------|

4. Öffnen Sie die Eingabemaske zur Konfiguration und der Auswahl der Elemente durch Klicken auf den Namen der Auswertungen in der Liste der Auswertungen.
5. Klicken Sie auf das Symbol zum Hinzufügen eines neuen Elements zur aktuellen Auswertung und wählen Sie anschließend als Typ des Elements "Letzte Messwerte".



Hinzufügen eines neuen Elements zur aktuellen Auswertung

1 neues Element zur Auswertung hinzufügen	3 Dropdown-Liste der verfügbaren Elemente
2 Eingabefenster zur Auswahl eines neuen Elements	4 Button "erstellen"

6. Konfigurieren Sie das Element "letzte Messwerte" wie folgt. Dadurch werden bei geöffneter Auswertung die letzten Messwerte der Messkanäle "Temperature", "Batery Voltage" und "USB Voltage" angezeigt. Dies ist vergleichbar mit der Anzeige der letzten Messwerte in der Liste der Auswertungen neben dem Namen der Auswertung. Allerdings können zu einer geöffneten Auswertung noch weitere Elemente, wie beispielsweise das im nächsten Schritt beschriebene Grafik-Element, hinzugefügt werden. Durch Klicken auf den Button "speichern" gelangen Sie wieder zur Übersicht der Elemente der aktuell geöffneten Auswertung zurück.

	Messstelle	Kanal	Typ	Min	Max
1	Wireless Thermometer	Temperature	Vertikale Skala	0	45
2	Wireless Thermometer	Battery Voltage	Runde Skala	0	5
3	Wireless Thermometer	USB Voltage	Ziffern		

support@microtronics.at abmelden support@microtronics.at
Hilfe zurück

– Letzte Messwerte 1

Name: Letzte Messwerte

Standard Sichtbarkeit: aufgeklappt

Größe: groß

1 Messstelle Wireless Thermometer Kanal Temperature Typ Vertikale Skala Min 0,0 Max 45,0
2 Messstelle Wireless Thermometer Kanal Battery Voltage Typ Runde Skala Min 0,0 Max 5,0
3 Messstelle Wireless Thermometer Kanal USB Voltage Typ Ziffern

Eingabemaske zur Konfiguration eines Elements des Typs "Letzte Messwerte"

1	frei wählbarer Name für das Auswerteelement
2	Auswahl der Messstelle. Das Element kann Daten von unterschiedlichen Messstellen beziehen. Es stehen alle Messstellen des aktuellen Kunden zur Verfügung.
3	Kanal der Messstelle, der angezeigt werden soll
4	Auswahl des Typs des zu verwendenden Anzeigeelements
5	unteres Skalenende des Anzeigeelements (nur für Skala und Balken-Elemente)
6	oberes Skalenende des Anzeigeelements (nur für Skala und Balken-Elemente)

7. Fügen Sie ein weiteres Element zur aktuellen Auswertung hinzu. Wählen Sie dieses Mal als Typ "Grafik".

8. Konfigurieren Sie das Element "Grafik" wie folgt. Dadurch wird bei geöffneter Auswertung der Verlauf der Messwerte der Messkanäle "Temperature", "Batery Voltage" und "USB Voltage" während der letzten 30min. angezeigt. Durch Klicken auf den Button "speichern" gelangen Sie wieder zur Übersicht der Elemente der aktuell geöffneten Auswertung zurück.

	Messstelle	Kanal	Auto Skala	Min	Max
1	Wireless Thermometer	Temperature	Null sichtbar	0	
2	Wireless Thermometer	Batery Voltage	aus	0	6
3	Wireless Thermometer	USB Voltage	aus	0	6

Eingabemaske zur Konfiguration eines Elements des Typs "Grafik"

1	frei wählbarer Name für das Auswerteelement
2	Zeitraum, der standardmäßig angezeigt werden soll Wählen Sie hier "30min.".
3	Auswahl der Messstelle. Das Element kann Daten von unterschiedlichen Messstellen beziehen. Es stehen alle Messstellen des aktuellen Kunden zur Verfügung.
4	Kanal der Messstelle, der angezeigt werden soll
5	Auswahl der zu verwendenden Skalierung
6	Bei "Auto Scale" "aus": unteres Skalenende Bei "Auto Scale" "Fließend" oder "Null sichtbar": Mindestskalengröße (Differenz zw. den Skalenenden)
7	oberes Skalenende

9. Aufgrund der Verbindungsart „online“ werden die Messwerte sofort nach deren Ermittlung zum Cloud-Server übermittelt und können mittels der in den vorherigen Schritten erstellten Auswertung visualisiert werden. Die Anzeige des Elements „Letzte Messwerte“ wird automatisch beim Empfang neuer Daten aktualisiert. Die Messwertgrafik des Elements „Grafik“ erfordert hingegen eine Userinteraktion (z.B. Klicken oder Verschieben der Messwertgrafik).



Visualisieren der Messdaten mittels Auswertung

- | | |
|---|--|
| 1 | Name der Auswertung, die zur Visualisierung der Messdaten herangezogen wird |
| 2 | schließt die Auswertung und kehrt zum Bereich "Messstellen / Applikationen" zurück |
| 3 | letzer Messwert des Messkanals "Temperature" |
| 4 | letzer Messwert des Messkanals "Battery Voltage" |
| 5 | letzer Messwert des Messkanals "USB Voltage" |
| 6 | Messwertgrafiken (Anzeige des Zeitverlaufs der Messwerte) |
| 7 | Checkbox zum Ein- und Ausblenden des Messkanals "Temperature" |
| 8 | Checkbox zum Ein- und Ausblenden des Messkanals "Battery Voltage" |
| 9 | Checkbox zum Ein- und Ausblenden des Messkanals "USB Voltage" |

2.4 Phase 4: Zugriff auf die Daten und Konfigurationen über die REST-API des Cloud-Servers mittels rapidM2M Playground

Wichtiger Hinweis: Bei der API handelt es sich um ein kostenpflichtiges Feature. Pro Messstelle/Applikation entstehen dadurch monatliche Kosten. Der Betrag wird immer am Monatsletzten berechnet und vom Guthaben abgezogen.

Diese Phase ist optional und dient dazu sich mit den durch die API des Cloud-Servers bereitgestellten Ressourcen und deren Handhabung vertraut zu machen.

1. Wechseln Sie zum Bereich "Meine Daten" und aktivieren Sie im Konfigurationsabschnitt "Grundeinstellungen" die Checkbox "API Verfügbar". Beim Klicken des Buttons "speichern" werden Sie nochmals darauf hingewiesen, dass es sich bei der API um ein kostenpflichtiges Feature handelt.

The screenshot shows a web-based configuration interface for a customer account. At the top, there's a header with the email 'support@microtronics.at', a log-out link 'abmelden support@microtronics.at', and navigation tabs: 'Messstellen / Applikationen', 'Messgeräte', 'Benutzer', 'Alarme', 'Statistik', and 'Meine Daten'. The 'Meine Daten' tab is highlighted with a yellow background. Below the tabs, there's a section titled 'Meine Daten' containing various input fields for customer information: Kunde*, Vorname, Nachname, Straße, PLZ, Ort, Firma, Tel, Umsatzsteuer-ID, Umsatzsteuer-ID verifiziert, and Zeitpunkt der Verifizierung. Underneath this is a section titled 'Kommentar'. At the bottom, there's a section titled 'Grundeinstellungen' with a checkbox labeled 'API verfügbar:'.

Übersicht des Bereichs "Meine Daten"

- | | |
|---|--|
| 1 | Schaltfläche für den Wechsel zum Bereich "Meine Daten" |
| 2 | Checkbox zum Aktivieren der API |

2. Wechseln Sie zurück zum Bereich "Messstellen / Applikationen" und klicken Sie auf "API", um den rapidM2M Playground zu öffnen. Der rapidM2M Playground ermöglicht es Ihnen, sich mit der API des Cloud-Servers vertraut zu machen und die bereitgestellten Funktionen zu testen.

Bereich "Messstellen / Applikationen", in dem alle zur Verfügung stehenden Messstellen angezeigt werden

1 öffnet den rapidM2M Playground

3. Geben Sie Ihre Zugangsdaten ein und wählen Sie das HTTP-Kommando "QUERY the youngest raw values" aus der Gruppe "Site's time series data". Dieses Kommando liefert den aktuellsten der gespeicherten Messdatensätze. Für den rapidM2M Playground können Sie die selben Zugangsdaten wie für den Cloud-Server verwenden.

Auswahl des HTTP-Kommandos "QUERY the youngest raw values"

- | | |
|---|---|
| 1 | Eingabefeld für den Benutzernamen |
| 2 | Auflistung der zur Verfügung stehenden HTTP-Kommandos. Die HTTP-Kommandos sind entsprechend ihrer Anwendungsbereiche gruppiert. |
| 3 | Eingabefeld für das Passwort |
| 4 | Fenster, in dem das gewählte HTTP-Kommando angezeigt wird |
| 5 | HTTP-Kommando zum Abfragen des aktuellsten der gespeicherten Messdatensätze |

4. Manche Ressource-Pfade von HTTP-Kommandos enthalten Wildcards (z.B. \$CID für den Kunden). Das aktuell gewählte HTTP-Kommando "QUERY the youngest raw values" benötigt die Angabe eines Kunden sowie einer Messstelle. Am Cloud-Server wird bei der Registrierung immer auch ein Kunde angelegt. Dieser erhält automatisch den selben Namen wie der Benutzer. Als Name für die Messstelle wird beim Anlegen automatisch die Seriennummer des Geräts verwendet. Der Name der Messstelle wurde allerdings im letzten Schritt der Phase 1 auf "Wireless Thermometer" geändert.

```
GET /1/customers/$CID/sites/$SID/histdata0/youngest
{
  "select": [
    "ch0",
    "ch3",
    "ch1"
  ]
}
```

Auswahl des Kunden sowie der Messstelle, die die Wildcards im Ressource-Pfad ersetzen sollen

- | | |
|----------|--|
| 1 | Auswahl des Kunden, der die Wildcard "\$CID" im Ressource-Pfad von HTTP-Kommandos ersetzt

Wählen Sie hier Ihren Benutzernamen. (Bei der Registrierung wurde ein Kunde, der den selben Name wie Ihr Benutzer trägt, angelegt.) |
| 2 | Auswahl der Messstelle, die die Wildcard "\$SID" im Ressource-Pfad von HTTP-Kommandos ersetzt

Wählen Sie hier "Wireless Thermometer". |

5. Um nun die Messdaten abrufen zu können, muss das HTTP-Kommando wie in der folgenden Abbildung dargestellt modifiziert werden. Für jeden Messwert, der gelesen werden soll, muss der im Schritt 10 der Phase 1 angegebene alternative Feldname (alias) verwendet werden. Achten Sie darauf, dass nach dem letzten Feldnamen kein Beistrich zu setzen ist. Klicken Sie danach auf den Button zum Ausführen des HTTP-Kommandos.

```
GET /1/customers/$CID/sites/$SID/histdata0/youngest
{
  "select": [
    "Temperature",
    "VoltageBattery",
    "VoltageUSB"
  ]
}
```

Modifizieren des HTTP-Kommandos "QUERY the youngest raw values"

- | | |
|----------|--|
| 1 | Messdatenkanal des Cloud-Servers, auf den zugegriffen werden soll

In Schritt 9 der Phase 1 wurde die Verwendung des Messdatenkanals o (histdatao) festgelegt. |
| 2 | Button zum Ausführen des HTTP-Kommandos |
| 3 | Liste der Messwerte des Messdatenkanals o, deren Werte abgerufen werden soll

In Schritt 10 der Phase 1 wurden die hier anzuführenden alternativen Feldnamen (alias) festgelegt. |
| 4 | Fenster, in dem das gewählte HTTP-Kommando angezeigt wird |

6. Im rechten Fenster wird die Antwort auf das HTTP-Kommando angezeigt. Das HTTP-Kommando "QUERY the youngest raw values" liest den zuletzt in der Datenbank des Cloud-Servers gespeicherten Messdatensatz aus. Die Werte entsprechen somit exakt jenen, die von den Anzeigeelementen der in Phase 3 erstellten Auswertung ausgegeben werden.

```
GET /1/customers/$CID/sites/$SID/histdata0/youngest
{
  "select": [
    "Temperature",
    "VoltageBattery",
    "VoltageUSB"
  ]
}
```

Query the youngest raw values of site's time series data (histdata0 to histdata9 available)

Annotations:

- 1 Fenster, in dem das JSON-Objekt angezeigt wird, das als Antwort auf das HTTP-Kommando erzeugt wird
- 2 Zeitstempel des Messdatensatzes
- 3 letzter Messwert des Messkanals "Temperature"
- 4 letzter Messwert des Messkanals "Battery Voltage"
- 5 letzter Messwert des Messkanals "USB Voltage"

Ergebnis des HTTP-Kommandos "QUERY the youngest raw values"

- | | |
|---|--|
| 1 | Fenster, in dem das JSON-Objekt angezeigt wird, das als Antwort auf das HTTP-Kommando erzeugt wird |
| 2 | Zeitstempel des Messdatensatzes |
| 3 | letzter Messwert des Messkanals "Temperature" |
| 4 | letzter Messwert des Messkanals "Battery Voltage" |
| 5 | letzter Messwert des Messkanals "USB Voltage" |
7. Für die Abfrage der Konfigurationen steht das HTTP-Kommando "QUERY a site's configuration" zur Verfügung. Da in Schritt 8 der Phase 1 die Entscheidung getroffen wurde, den Konfigurationsblock o (config0) zu verwenden, ist keine Modifikation des HTTP-Kommandos erforderlich. Klicken Sie also direkt nach der Auswahl des HTTP-Kommandos auf den Button zum Ausführen des HTTP-Kommandos.

support@microtronics.at SCID SSID

GET /1/customers/\$CID/sites/\$SID/config0

Annotations:

- 1 Konfigurationsblock, auf den zugegriffen werden soll
- 2 Button zum Ausführen des HTTP-Kommandos
- 3 HTTP-Kommando zum Abfragen eines der Konfigurationsblöcke
- 4 Fenster, in dem das gewählte HTTP-Kommando angezeigt wird

Note:
The number of "fieldX" values returned, and their type depend on the CONNECTOR settings assigned to this site.

Auswahl des HTTP-Kommandos "QUERY the youngest raw values"

- | | |
|---|--|
| 1 | Konfigurationsblock, auf den zugegriffen werden soll |
| | In Schritt 8 der Phase 1 wurde die Verwendung des Konfigurationsblocks o (config0) festgelegt. |
| 2 | Button zum Ausführen des HTTP-Kommandos |
| 3 | HTTP-Kommando zum Abfragen eines der Konfigurationsblöcke |
| 4 | Fenster, in dem das gewählte HTTP-Kommando angezeigt wird |

8. Im rechten Fenster wird die Antwort auf das HTTP-Kommando angezeigt.

The screenshot shows the rapidM2M Playground interface. On the left, there's a sidebar with navigation links like 'The first thing upon user login', 'Manage a customer', 'Manage a site', 'Manage a device', and 'Site's status & configuration data'. The main area has a header with 'support@microtronics.at', '*****', 'support@microtronics.at', and 'Wireless Thermometer'. Below this is a search bar with 'GET /1/customers/\$CID/sites/\$SID/config0'. To the right, there's a large text box containing a JSON response. Above the text box, a status bar shows '200' and '165 ms'. The JSON response is annotated with numbers 1 through 5 pointing to specific fields:

```

1 {
  "stamp": "20170517070652270",
  "recordinterval": 10,
  "transmissioninterval": 60,
  "transmissionmode": 2
}
  
```

Annotations:

- 1**: Zeitstempel des Messdatensatzes (Stamp of the measurement data set)
- 2**: zeitlicher Abstand der Übertragungen (zeitlicher Abstand der Übertragungen)
- 3**: zeitlicher Abstand der Messdatenaufzeichnungen (zeitlicher Abstand der Messdatenaufzeichnungen)
- 4**: Auswahl der Verbindungsart (Auswahl der Verbindungsart)
- 5**: Zeitstempel des Messdatensatzes (Stamp of the measurement data set)

Ergebnis des HTTP-Kommandos "QUERY a site's configuration"

- | | |
|----------|---|
| 1 | Fenster, in dem das JSON-Objekt angezeigt wird, das als Antwort auf das HTTP-Kommando erzeugt wird |
| 2 | Zeitstempel des Messdatensatzes |
| 3 | zeitlicher Abstand der Messdatenaufzeichnungen |
| 4 | zeitlicher Abstand der Übertragungen (nur relevant für die Verbindungsarten "Intervall" und "Wakeup") |
| 5 | Auswahl der Verbindungsart (0 = Intervall, 1 = Wakeup, 2 = Online) |

9. Mittels des HTTP-Kommandos "MODIFY a site's configuration" können die Werte der Konfigurationsparameter angepasst werden. Im konkretem Beispiel soll die Verbindungsart von "Online" auf "Wakeup" geändert werden. Um einen Parameter modifizieren zu können muss dessen alternativer Feldname (alias) bekannt sein. Es ist sowohl möglich einen einzelnen als auch mehrere Konfigurationsparameter mit einem Aufruf des HTTP-Kommandos "MODIFY a site's configuration" zu verändern. Achten Sie darauf, dass nach dem letzten Feldnamen kein Beistrich zu setzen ist. Modifizieren Sie das HTTP-Kommando nach der Auswahl wie in der folgenden Abbildung dargestellt und klicken Sie anschließend auf den Button zum Ausführen des HTTP-Kommandos.

The screenshot shows the rapidM2M Playground interface. On the left, there is a sidebar with user information (support@microtronics.at, password placeholder) and a navigation menu. The main area displays an API call:

```
PUT /1/customers/$CID/sites/$SID/config0
{
  "transmissionmode": 1
}
```

Annotations with numbers 1 through 5 point to specific elements:

- 1: The 'config0' part of the URL.
- 2: The execute button (play icon).
- 3: The 'transmissionmode' key in the JSON body.
- 4: The value '1' assigned to 'transmissionmode'.
- 5: The 'MODIFY a site's configuration' button in the sidebar.

Below the code editor, a note states: "Modify a site's configuration (config0 to config9 available)." and "Note: Omit properties from the PUT if you do not want to modify them!"

Modifizieren des HTTP-Kommandos "MODIFY a site's configuration"

1 Konfigurationsblock, auf den zugegriffen werden soll

In Schritt 8 der Phase 1 wurde die Verwendung des Konfigurationsblocks o (config0) festgelegt.

2 Button zum Ausführen des HTTP-Kommandos

3 alternativer Feldname (alias) des Parameters im Konfigurationsblock o, der modifiziert werden soll

In Schritt 8 der Phase 1 wurde der hier anzuführende alternative Feldname (alias) festgelegt.

4 Wert der gesetzt werden soll

Wählen Sie hier "1" für die Verbindungsart "Wakeup". Das Gerät meldet sich im Übertragungsintervall. Es kann jedoch auch eine Verbindung durch den Server ausgelöst werden.



5 HTTP-Kommando zum Modifizieren eines der Konfigurationsblöcke

10. Da das HTTP-Kommando "MODIFY a site's configuration" keinen Response Body liefert, ist das Fenster, in dem das JSON-Objekt angezeigt wird, leer. Die erfolgreiche Ausführung des HTTP-Kommandos lässt sich anhand des Response-Code 204 im rechten Bereich erkennen.

The screenshot shows the rapidM2M Playground interface after executing the command. The main area displays the same API call as before, but the response window on the right now contains the status code:

```
204
```

Annotations with numbers 1 and 2 point to the status code and the response time (94 ms) respectively.

Ergebnis des HTTP-Kommandos "MODIFY a site's configuration"

1 Response-Code, der vom Cloud-Server als Antwort auf das HTTP-Kommando gesendet wurde

2 Fenster, in dem das JSON-Objekt angezeigt wird, das als Antwort auf das HTTP-Kommando erzeugt wird

11. Aufgrund der Verbindungsart "online" wird die über das HTTP-Kommando "MODIFY a site's configuration" vorgenommene Konfigurationsänderung sofort zum rapidM2M M3 übermittelt. Im konkreten Fall bewirkt die Änderung, dass das Gerät in den Verbindungsmodus "Wakeup" übergeht. Erkennbar ist dies einerseits am Erlöschen der LED und andererseits daran, dass in der Liste der Messstellen / Applikationen die rotierenden Pfeile ausgeblendet werden. Weiters wird in der Liste der Messstellen / Applikationen das Symbol zum Versenden einer Wakeup-SMS eingeblendet. Beim Empfang einer solchen SMS baut das Gerät sofort eine Verbindung zum Server auf.

Liste der Messstellen / Applikationen

- 1 versendet eine Wakeup-SMS an das rapidM2M M3 , um dieses anzuweisen eine sofortige Verbindung zum Cloud herzustellen

Wichtiger Hinweis: Durch den Versand einer Wakeup-SMS entstehen Kosten.

12. Zur Fortführung dieses Tutorial ist es erforderlich das rapidM2M M3 wieder in den Verbindungsmodus "Online" zu versetzen. Verwenden Sie hierfür erneut das HTTP-Kommando "MODIFY a site's configuration". Modifizieren Sie das HTTP-Kommando nach der Auswahl wie in der folgenden Abbildung dargestellt und klicken Sie anschließend auf den Button zum Ausführen des HTTP-Kommandos.

Modifizieren des HTTP-Kommandos "MODIFY a site's configuration"

- | | |
|--|--|
| 1 | Konfigurationsblock, auf den zugegriffen werden soll |
| 2 | Button zum Ausführen des HTTP-Kommandos |
| 3 | alternativer Feldname (alias) des Parameters im Konfigurationsblock o, der modifiziert werden soll |
| 4 | Wert der gesetzt werden soll |
| Wählen Sie hier "1" für die Verbindungsart "Online". Das Gerät trennt die Verbindung nicht und übermittelt kontinuierlich die Messdaten. | |
| 5 | HTTP-Kommando zum Modifizieren eines der Konfigurationsblöcke |

13. Die erfolgreiche Ausführung des HTTP-Kommandos lässt sich wiederum anhand des Response-Code 204 im rechten Bereich erkennen.

14. Im Verbindungsmodus "Wakeup" besteht keine aktive Kommunikation zwischen dem Cloud-Server und dem rapidM2M M3 . Die mittels des HTTP-Kommandos "MODIFY a site's configuration" vorgenommene Konfigurationsänderung wurde somit noch nicht zum Gerät übermittelt. Drücken Sie daher die Taste des rapidM2M M3 , um einen Verbindungsaufbau mit anschließender Synchronisation der Konfigurationen auszulösen.
15. Wechseln Sie zum Cloud-Server und warten Sie bis in der Liste der Messstellen / Applikationen angezeigt wird, dass das rapidM2M M3 mit dem Cloud-Server verbunden ist (rotierende Pfeile). Dies kann bis zu 2 Minuten dauern.

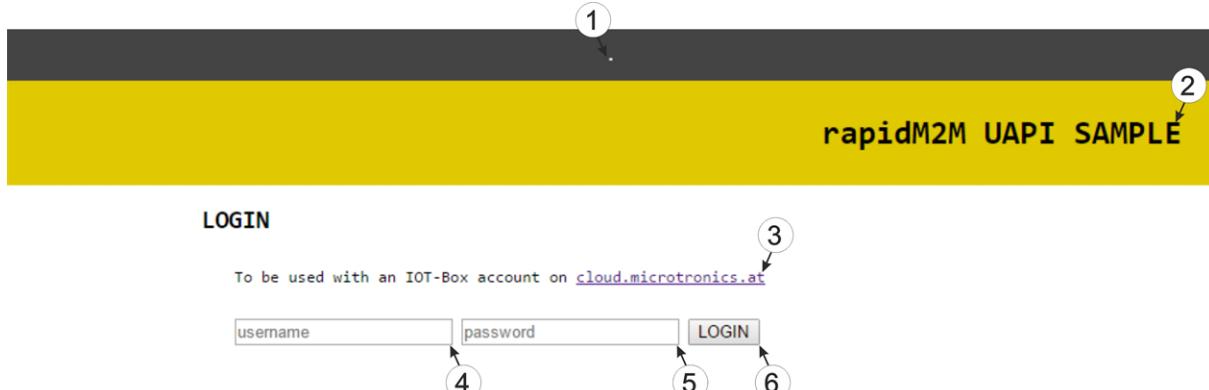


Liste der Messstellen / Applikationen

- | | |
|----------|---|
| 1 | Zeigt an, dass das rapidM2M M3 mit dem Server verbunden ist und Daten überträgt. Dieses Symbol wird ausgeblendet, wenn die Verbindung getrennt wird. |
| 2 | Informationen zum Zeitpunkt der Kommunikation zwischen Messgerät und Server <ul style="list-style-type: none"> Messgerät ist mit dem Server verbunden und überträgt Daten: Zeitpunkt des letzten Verbindungsbaus Messgerät ist momentan nicht mit dem Server verbunden: Zeitpunkt des letzten Verbindungsabbaus |
| 3 | Das Symbol gibt die Signalstärke an. Ein Klick auf das Symbol zeigt Informationen zur Funkzelle, die bei der letzten Verbindung verwendet wurde sowie zu den bei der letzten Positionsbestimmung berücksichtigten Funkzellen an. |

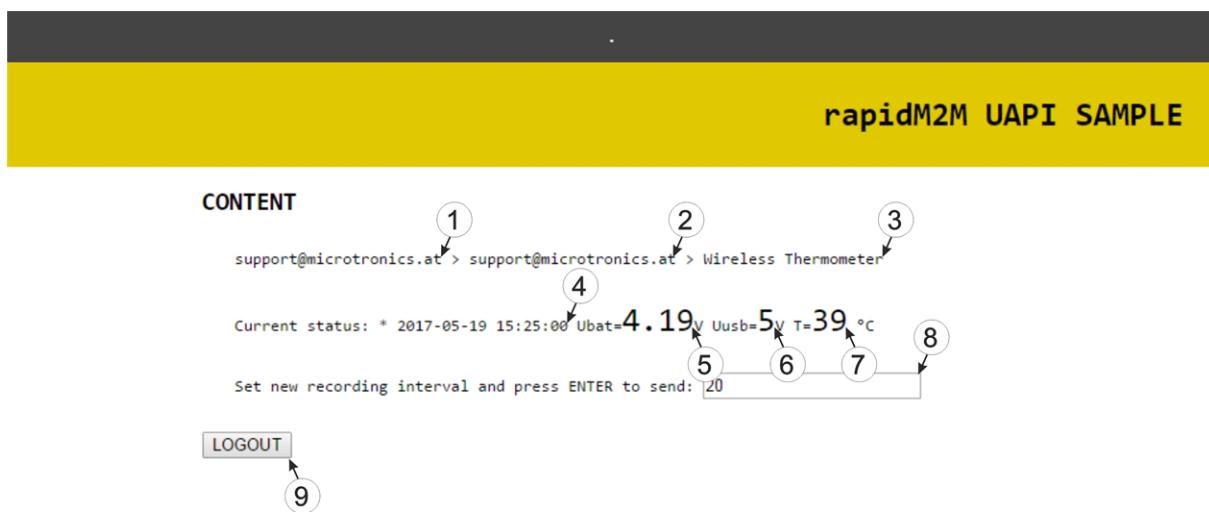
2.5 Phase 5: Erstellen eines eigenen Dashboards

Hinweis: Manche der verwendeten Funktionen werden nicht von allen Browsern unterstützt. Daher wird die Verwendung von "Google Chrome" ab Version 58 empfohlen.



Login-Page des Dashboards

1 Statusanzeige	4 Eingabefeld für den Benutzernamen (E-Mail-Adresse)
2 Name des Dashboards	5 Eingabefeld für das Passwort
3 Web-Adresse des Cloud-Servers	6 Prüft die User Credentials und wechselt zur Content-Page



Content-Page des Dashboards

1 Name des Kunden (E-Mail-Adresse, die bei der Registrierung des Accounts angegeben wurde)	6 letzter Messwert des Messkanals "USB Voltage"
2 Benutzername (E-Mail-Adresse)	7 letzter Messwert des Messkanals "Temperature"
3 Name der Messstelle	8 Eingabefeld für den zeitlichen Abstand der Messdatenaufzeichnungen (wird bei Drücken der "Enter"-Taste übernommen)
4 Zeitstempel des Messdatensatzes	9 meldet den aktiven Benutzer ab und wechselt zur Login-Page
5 letzter Messwert des Messkanals "Battery Voltage"	

Der erste Schritt ist nur erforderlich, wenn Sie die Phase 4 dieses Tutorials übersprungen haben.

- Wechseln Sie zum Bereich "Meine Daten" und aktivieren Sie im Konfigurationsabschnitt "Grundeinstellungen" die Checkbox "API Verfügbar". Beim Klicken des Buttons "speichern" werden Sie nochmals darauf hingewiesen, dass es sich bei der API um ein kostenpflichtiges Feature handelt.

The screenshot shows a configuration interface for 'Meine Daten'. At the top, there's a header with the email 'support@microtronics.at', a log-out link 'abmelden support@microtronics.at', and navigation tabs: 'Messstellen / Applikationen', 'Messgeräte', 'Benutzer', 'Alarme', 'Statistik', and 'Meine Daten' (which is highlighted with a yellow background). Below the tabs is a toolbar with 'API Hilfe'. The main area contains several input fields for company information: 'Kunde*' (support@microtronics.at), 'Vorname' (Support), 'Nachname' (Microtronics), 'Straße' (Hauptstrasse 7), 'PLZ' (3244), 'Ort' (Ruprechtshofen), 'Firma' (Microtronics Engineering GmbH), 'Tel.' (+4327567718023), 'Umsatzsteuer-ID' (empty), 'Umsatzsteuer-ID verifiziert' (nicht benötigt), and 'Zeitpunkt der Verifizierung' (empty). There are also sections for '+ Kommentar' and '- Grundeinstellungen' which includes the 'API verfügbar' checkbox (marked with a yellow circle '2').

Übersicht des Bereichs "Meine Daten"

- | | |
|----------|--|
| 1 | Schaltfläche für den Wechsel zum Bereich "Meine Daten" |
| 2 | Checkbox zum Aktivieren der API |

- Legen Sie eine neue leere html-Datei mit UTF-8 Kodierung an.

Hinweis: Wollen Sie die schrittweise Erstellung des Dashboards überspringen, öffnen Sie anstelle der Erstellung einer neuen html-Datei die Datei "Dashboard.html" mit einem Browser und setzen Sie das Tutorial bei Schritt 11 fort. Die Datei "Dashboard.html" ist im Paket der ergänzenden Dateien zu diesem Tutorial enthalten.

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title> </title>
  </head>

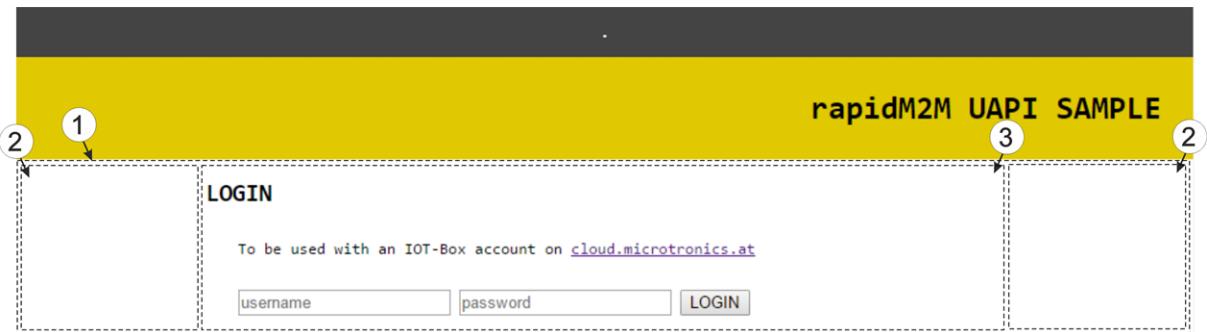
  <body>

  </body>
</html>
```

3. Zunächst gilt es den Header der html-Datei zu erstellen. Neben dem Titel der Datei enthält er die Formatanweisungen für die Tags und Klassen.

```
<head>
  <!-- Zeichencodierung auf UTF-8 setzen -->
  <meta charset="utf-8">
  <!-- Titel der Datei, der unter anderem in der Titelleiste des Browserfensters angezeigt wird -->
  <title>rapidM2M UAPI SAMPLE for User Agents</title>
  <style>
    /* Body: gesamte Breite des Browsersfensters nutzen, Schriftart "Monospace", kein Außenabstand */
    body { width:100%;font-family:Monospace;margin:0}
    /* Dashboard-Name: kein Außenabst., gelber Hintergrund, rechtsbündig, Innenabst. = 1x Schriftgr. */
    h1 { margin:0;background: #e0c900;text-align:right;padding:1em}
    /* Master: aktiviert die Verwendung der Flex-Eigenschaften für die Unterelemente dieses Bereichs */
    .master{ display:flex }
    /* Spacer: Der Bereich nimmt mit dem Faktor 1 den verfügbaren verbleibenden Raum eines Bereichs ein */
    .spacer{ flex-grow:1 }
    /* Login-Page bzw. Content-Page: bei Initialisierung ausgeblendet, Breite = 50x Schriftgröße */
    .page { display: none; width:50em}
    /* Alle DIV's innerhalb der Login-Page bzw. Content-Page: Außenabstand = 2x Schriftgr. */
    .page>div{ margin:2em }
    /* Statusanzeige: weiße Schrift auf schwarzem Hintergrund, zentriert, Innenabst. = 1xSchriftgr. */
    #uapi_msg{ background:#444;color:white;text-align:center;padding:1em}
    /* Messwertanzeige: Schriftgröße = 2x Content-Page Standard-Schriftgröße, normaler Schriftstil */
    em { font-size:2em;font-style:normal; }
  </style>
</head>
```

Hinweis: Um den Inhalt der Login-Page bzw. Content-Page in der Mitte des Browserfensters darzustellen, wird zunächst ein Bereich der Klasse "master" angelegt. Für diesen wird die Verwendung der Flex-Eigenschaften für dessen Unterelemente aktiviert. Als Unterelement des "master" Bereichs werden der Reihe nach ein Bereich der Klasse "spacer", zwei Bereiche der Klasse "page" (für Login-Page und Content-Page, wobei immer nur eine der beiden eingeblendet wird) und abschließend wieder ein Bereich der Klasse "spacer" angelegt. Für die Klasse "page" wurde eine feste Breite definiert. Mittels der Eigenschaft "flex-grow:1" wurde für die Klasse "spacer" festgelegt, dass der verbleibende, verfügbare Raum innerhalb des Bereichs "master" zwischen den beiden Bereichen der Klasse "page" gleichmäßig aufgeteilt wird. Dadurch wird der Inhalt der Login-Page bzw. Content-Page immer automatisch mittig ausgerichtet, auch wenn die Größe des Browserfensters verändert wird.



Ergänzende Erklärung zum Aufbau des Anzeigebereichs des Dashboards

1 Bereich der Klasse "master"	3 Bereich der Klasse "page"
2 Bereich der Klasse "spacer"	

4. Als nächstes ist das User-Interface des Dashboards zu erzeugen. Dieses befindet sich zusammen mit dem JavaScript-Code im Body-Abschnitt der html-Datei. Zur besseren Übersicht wurde die Erstellung des JavaScript-Codes auf die nächsten 5 Schritte aufgeteilt. Der Code ist unter der Kommentarzeile "`<!-- JavaScript-Code (siehe Schritt 5 bis 9) -->`" einzufügen.

Hinweis: Bitte beachten Sie, dass sowohl die Login-Page als auch die Content-Page aufgrund der Definition der Klasse "page" zunächst ausgeblendet sind und erst durch den JavaScript-Code situationsabhängig eingeblendet werden.

```

<body>
  <div id="uapi_msg"></div>          <!-- Statusanzeige -->

  <h1>rapidM2M UAPI SAMPLE</h1>      <!-- Name des Dashboards -->
  <div class="master">                  <!-- für mittige Ausrichtung bei Größenänderung des Fensters -->
    <div class="spacer"></div>          <!-- Spacer links von der Login-Page bzw. Content-Page -->
    <div id="pg_login" class="page">    <!-- Login-Page -->
      <h2>LOGIN</h2>                <!-- Titel der Login-Page -->
      <div>                          <!-- Anzeige der Web-Adresse des Servers -->
        To be used with an IOT-Box account on
        <a href="https://cloud.microtronics.at">cloud.microtronics.at</a>
      </div>
      <div>                          <!-- Eingabefeld für den Benutzernamen, Hinweistext auf "username" setzen, bekommt Focus -->
        <input id="login_usr" type="text" placeholder="username" autofocus></input>
      <!-- Eingabefeld für das Passwort, Hinweistext auf "password" setzen -->
        <input id="login_pwd" type="password" placeholder="password"></input>
      <!-- Login-Button zum Prüfen der User Credentials und Wechsel zur Content-Page -->
        <button id="btn_login">LOGIN</button>
      </div>
    </div>
  </div>

  <div id="pg_content" class="page"> <!-- Content-Page -->
    <h2>CONTENT</h2>                <!-- Titel der Content-Page -->
    <div id="me">loading...</div>      <!-- Anzeige von Kunden-, Benutzer- u. Messstellennamen -->
    <div>                          <!-- Anzeige der letzten Messwerte (inkl. Zeitstempel) -->
      Current status: <span id="status">loading...</span>
    </div>
    <div>                          <!-- Eingabefeld für das Aufzeichnungsintv., Hinweistext "new value + ENTER", bekommt Focus -->
      Set new recording interval and press ENTER to send:
      <input id="ed_reciv" type="text" placeholder="new value + ENTER" autofocus></input>
    </div>

    <!-- Button zum Abmelden des aktiven Benutzers und Wechsel zur Login-Page -->
    <button id="btn_logout">LOGOUT</button>
  </div>
  <div class="spacer"></div>          <!-- Spacer rechts von der Login-Page bzw. Content-Page -->
</div>

<!-- Einbinden der jQuery-Library -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<!-- Einbinden der rapidM2M UAPI Library -->
<script src="https://cdn.microtronics.at/libs/fe/rapidm2m-uapi.js"></script>
<script>

  <!-- JavaScript-Code (siehe Schritt 5 bis 9) -->

</script>
</body>

```

5. Nun sind einige Konstanten und globale Variablen sowie zwei einfache Funktionen für den Programmablauf zu definieren.

```

"use strict";                                // Der JavaScript-Code soll im "strict mode" ausgeführt werden.

showpage( '#pg_login');                     // Login-Page einblenden

// Konstante, die die Web-Adresse des Servers enthält
const MY_HOST = 'https://cloud.microtronics.at/api';

/*=====
Funktion zum Aus-/Einblenden der Login-Page und Content-Page

pg:[in]
  id der Page, die eingeblendet werden soll
=====*/
function showpage(pg) {
    $('.page').hide();                      // Alle Pages ausblenden
    $(pg).show();                          // Page, deren Id übergeben wurde, einblenden
    // Sucht in der Page ein Objekt mit gesetztem "autofocus"-Attribut u. setzt den Focus auf dieses Obj.
    $(pg).find('[autofocus]').focus();
}

let uapi;          // Variable für eine neue Instanz der JavaScript-Klasse "UAPI"
let poll_tmr;      // Variable zur Aufnahme der von der Funktion "setTimeout" zurückgelieferten ID
let poll_spin=0;   // Variable zum Toggeln des Symbols zur Anzeige der Aktivität in der Content-Page

// Konstante für das js-Objekt das auf das DOM Element des Eingabefelds "Aufzeichnungsitz." verweist
const $ed_reciv= $('#ed_reciv');

/*=====
Zeigt im Bereich der Statusanzeige den übergebenen String an. Wenn ein Leerstring oder
kein String übergeben wird, wird in der Statusanzeige das Zeichen "." angezeigt.

msg:[in]
  String der im Bereich der Statusanzeige angezeigt werden soll
=====*/
function msg(msg) {
    $('#uapi_msg').html( msg || '.');
}

```

6. Im Anschluss gilt es die Funktion, die im Sekundenintervall die Anzeige der letzten Messwerte aktualisiert, zu erstellen.

```
=====
Aktualisiert den Bereich in dem die letzten Messwerte angezeigt werden
Nach dem Absetzen der API-Abfrage setzt diese Funktion einen Timeout von 1sec. nach dessen
Ablauf sie wieder aufgerufen wird. Der erste Aufruf dieser Funktion erfolgt nach
erfolgreicher Überprüfung der User Credentials beim Klicken des Login-Buttons
=====*/
function poll_status(){

//Konstante für die Liste der Messwerte (alias), deren akt. Werte vom Server gelesen werden sollen
const o = {
    select:["VoltageBattery","VoltageUSB","Temperature"]
};

/* Get Zugriff auf den aktuellsten der gespeicherten Messdatensätze des Messdatenkanals 0, wobei die
in der Konstante "o" angegebenen Messwerte gelesen werden. Die anonyme Funktion verarbeitet den
Response-Code (err) und das zurückgelieferte JSON-Array(x) */
uapi.get( '1/customers/$cid/sites/$sid/histdata0/youngest', o, (err,x)=>{

/* Variable zur Aufnahme des Strings der im Bereich für Anzeige der letzten Messwerte angezeigt wird.
Das 1ste Zeichen wird bei jedem Aufruf abwechselnd auf "*" o. "." gesetzt, um Aktivität anzuzeigen*/
let s = (++poll_spin & 1) ? '*' :'.';

if (err) // Wenn ein Fehler aufgetreten ist ->
    s += err // Fehlercode zum String "s" hinzufügen
else if (!x.length) // Andernfalls -> Wenn JSON-Array die Länge 0 hat ->
    s += '(no data yet available)';//Hinweis, dass keine Daten vorhanden sind, zum String "s" hinzufügen
else {
    /* Den Zeitstempel aus dem JSON-Arrayelement [0][0] in das Format "yyyy-mm-dd hh:mm:ss[.zzz]" umwandeln, den "[.zzz]"-Anteil entfernen und das Ergebnis zum String "s" hinzufügen */
    s += uapi.stampFormat( x[0][0]).substr(0,19);
    /* Die Messwerte aus dem JSON-Array inkl. entsprechender Beschriftung zum String "s" hinzufügen
    s += ` Ubat=<em>${x[0][1]}</em>V Uusb=<em>${x[0][2]}</em>V T=<em>${x[0][3]}</em> &deg;C`;
}

$( '#status' ).html( s); // Im Bereich für die letzten Messwerte den String "s" anzeigen

/* Timeout von 1000ms setzen und zurückgelieferte ID in Variable speichern. Nach Ablauf des Timeouts
wird die Funktion "poll_status" erneut aufgerufen.
poll_tmr= setTimeout( poll_status, 1000);
});;

}

```

7. Jetzt muss festgelegt werden welche Aktionen beim Klicken des Login-Buttons ausgeführt werden.

```

=====
Beim Klicken auf den Login-Button werden zunächst die User Credentials geprüft und die
erste Messstelle innerhalb des ersten Kunden für den der Benutzer die Zugriffsberechtigung
besitzt ausgewählt. Anschließend wird das aktuelle Aufzeichnungsintervall vom Server
gelesen, zur Content-Page gewechselt und die Funktion zum Aktualisieren des Bereichs mit
den letzten Messwerten das erste Mal aufgerufen. Dieser Funktion ruft sich danach durch
Setzen des Timeouts im Sekundenintervall selbst immer wieder auf.
=====

$('#btn_login').click(()=>{

    msg('connecting...');           // Im Bereich der Statusanzeige den Text "connecting..." anzeigen

    uapi=new UAPI(                // Eine neue Instanz der JavaScript-Klasse "UAPI" anlegen und
        $('#login_usr').val(),     // den Benutzernamen aus dem Eingabefeld in die Klasse übernehmen
        $('#login_pwd').val(),     // das Passwort aus dem Eingabefeld in die Klasse übernehmen
        MY_HOST);                  // Web-Adresse des Servers von der Konstante in die Klasse übernehmen

    // Globale Fehlersignalisierung festlegen
    uapi.onError= (err,info)=>{      // Callback in die Instanz der JavaScript-Klasse "UAPI" übernehmen
        switch(err){               // Switch des Response-Code (err) ->
            // ungültige User Credentials -> In der Statusanzeige den Text "Illegal user credentials!" anzeigen
            case 401: msg('Illegal user credentials!'); break;
            // bei allen anderen Fehlern -> Response-Code und zusätzliche Infos in der Statusanzeige anzeigen
            default : msg('ERR#'+err +' ... ' + info);
        }
    }

    /* Liste der Kunden ermitteln, auf die der akt. Benutzer Zugriffsrechte besitzt. Die anonyme Funktion
       verarbeitet den Response-Code (err) und das zurückgelieferte JSON-Objekt(me) mit der Kundenliste */
    uapi.get('1/me',(err,me)=>{

        if (err) return;           // bei Fehler auf der Login-Page bleiben
                                    // (Restliche Behandlung übernimmt die globale Fehlersignalisierung.)

        /* Ersten Kunden aus der Liste wählen. In den folgenden Ressourcen-Pfaden wird "$cid" durch den in
           die Instanz "uapi" übernommenen Kundennamen ersetzt.
           uapi.placeholders.$cid= me.customers[0]; */

        /* Liste der Messstellen, die dem gewählten Kunden zugewiesen sind, ermitteln. Die anonyme Funktion
           verarbeitet den Response-Code (err) und das JSON-Objekt(sites) mit der Messstellenliste */
        uapi.get('1/customers/$cid/sites',(err,sites)=>{

            if (err) return;           // bei Fehler auf der Login-Page bleiben
                                    // (Restliche Behandlung übernimmt die globale Fehlersignalisierung.)

            /* Erste Messstelle aus der Liste wählen. In den folgenden Ressourcen-Pfaden wird "$sid" durch die
               in die Instanz "uapi" übernommene unique ID der Messstelle ersetzt.
               uapi.placeholders.$sid= sites[0]._uid; */

            /* Konfig 0 der gewählten Messstelle auslesen. Die anonyme Funktion verarbeitet den Response-Code
               (err) und das JSON-Objekt(x) mit den Konfigurationsparametern.
               uapi.get('1/customers/$cid/sites/$sid/config0',(err,x)=>{

                    if (err) return;           // bei Fehler auf der Login-Page bleiben
                                    // (Restliche Behandlung übernimmt die globale Fehlersignalisierung.)

                    // Vom Server gelesenes Aufzeichnungsintv. in das Eingabefeld für das Aufzeichnungsintv schreiben
                    $ed_reciv.val( x.recordinterval);

                    msg();                   //Bereich der Statusanzeige löschen

                    showpage('#pg_content'); //Content-Page einblenden und gleichzeitig Login-Page ausblenden

                    //Aktuell gewählten Kunden-, Benutzer- u. Messstellennamen im entsprechenden Bereich anzeigen
                    $('#me').text(
                        me.user.email + ' > ' +
                        me.customers[0] + ' > ' +
                        sites[0].name);

                    //Funktion zum Aktualisieren des Bereichs mit den letzten Messwerten das erste Mal aufrufen
                    poll_status();
                });
            });
        });
    });
});

```

8. Danach gilt es zu regeln in welcher Form die Eingabe eines geänderten Aufzeichnungsintervalls behandelt wird.

```
=====
Steht der Cursor im Eingabefeld für das Aufzeichnungsintervall und wird die ENTER-Taste gedrückt, wird der in das Eingabefeld eingegebene Wert in die Konfig 0 übernommen und zum Server übermittelt. Dabei wird auch sichergestellt, dass das Mindestintervall eingehalten wird.
=====
//Wenn eine Taste losgelassen wurde, verarbeitet die anonyme Funktion das Eventobjekt
$ed_reciv.on('keyup', (ev)=>{

    if (ev.keyCode !== 13) return; // Wenn es sich nicht um die ENTER-Taste handelt -> return

    msg('Saving...'); // Im Bereich der Statusanzeige den Text "Saving..." anzeigen

    //Konstante die sowohl den Parameter (alias) als auch den zu setzenden Wert angibt
    const o= {
        //bei ungültigen Eingaben bzw. Werten < 10sec. das Aufzeichnungsintv. auf 10sec. setzen
        recordinterval: Math.max( $ed_reciv.val() || 10, 10)
    }

    /* Mittels PUT die Konfig 0 der beim Login gewählten Messstelle aktualisieren, wobei die Konstante "o" sowohl den Parameter als auch den zu setzenden Wert angibt. Die anonyme Funktion verarbeitet den Response-Code (err)
    */
    uapi.put( '/customers/$cid/sites/$sid/config0', o, (err)=>{

        if (err) return;
        // bei Fehler -> return
        // (Restliche Behandlung übernimmt die globale Fehlersignalisierung.)

        msg();
        //Bereich der Statusanzeige löschen
    });
});
});
```

9. Zuletzt ist es noch erforderlich zu definieren wie das Dashboard beim Klicken auf den Logout-Button reagieren soll.

```
=====
Beim Klicken des Logout-Buttons wird die sekündliche Aktualisierung des Bereichs mit den letzten Messwerten gestoppt, die Instanz der JavaScript-Klasse "UAPI" welche die aktuellen User Credentials enthält, gelöscht und die Login-Page eingeblendet.
=====
$('#btn_logout').click(()=>{

    /* Durch Löschen des Timeouts wird die Schleife in der sich die Funktion zum Aktualisieren des Bereichs mit den letzten Messwerten permanent selbst wieder aufruft, unterbrochen. */
    clearTimeout( poll_tmr );

    uapi= null;
    showpage('#pg_login');
    // Instanz der Klasse "UAPI" mit den User Credentials löschen
    // Login-Page einblenden und gleichzeitig Content-Page ausblenden
});
```

10. Speichern Sie die soeben erstellte Datei und öffnen Sie sie anschließend mit einem Browser.

11. Geben Sie Ihren Benutzername (E-Mail-Adresse) und Ihr Passwort in die dafür vorgesehenen Felder ein und klicken Sie auf den Login-Button.



Login-Page des Dashboards

1 Eingabefeld für den Benutzernamen (E-Mail-Adresse)	3 Prüft die User Credentials und wechselt zur Content-Page
2 Eingabefeld für das Passwort	

12. Die nun geöffnete Content-Page zeigt nun die zuletzt ermittelten Messwerte sowie den Zeitpunkt zu dem die Messwerte erzeugt wurden an. Wenn Sie den in diesem Tutorial empfohlenen Wert von 10sec. verwenden, springt der Zeitpunkt der Messung immer um eben diese 10sec. weiter.

The screenshot shows a Content Page titled "rapidM2M UAPI SAMPLE". The page displays the following information:

- User path: support@microtronics.at > support@microtronics.at > Wireless Thermometer
- Current status: * 2017-05-19 15:25:00
- Metric values: Ubatt=4.19V, Uusb=5V, T=39 °C
- Text input field: Set new recording interval and press ENTER to send:
- Logout button: LOGOUT

Annotations with numbers 1 through 4 point to specific elements: 1 points to the timestamp; 2 points to the battery voltage; 3 points to the USB voltage; and 4 points to the temperature.

Content-Page des Dashboards

1 Zeitstempel des Messdatensatzes	3 letzter Messwert des Messkanals "USB Voltage"
2 letzter Messwert des Messkanals "Battery Voltage"	4 letzter Messwert des Messkanals "Temperature"

13. Ändern Sie das Aufzeichnungsintervall auf 15 und verlassen Sie das Eingabefeld durch Drücken der ENTER-Taste. Die erfolgreiche Übernahme des neuen Aufzeichnungsintervalls lässt sich daran erkennen, dass nun der Zeitpunkt der Messung jeweils um 15sec. weiterspringt.

The screenshot shows a Content Page titled "rapidM2M UAPI SAMPLE". The page displays the following information:

- User path: support@microtronics.at > support@microtronics.at > Wireless Thermometer
- Current status: * 2017-05-19 15:25:00
- Metric values: Ubatt=4.19V, Uusb=5V, T=39 °C
- Text input field: Set new recording interval and press ENTER to send:
- Logout button: LOGOUT

Annotations with numbers 1 and 2 point to specific elements: 1 points to the timestamp; 2 points to the text input field.

Content-Page des Dashboards

1 Zeitstempel des Messdatensatzes	2 Eingabefeld für den zeitlichen Abstand der Messdataufzeichnungen (wird bei Drücken der "Enter"-Taste übernommen)
-----------------------------------	--

Kontaktinformationen

Support & Service:

Microtronics Engineering GmbH

Hauptstrasse 7

3244 Ruprechtshofen

Austria, Europe

Tel. +43 (0)2756 7718023

support@microtronics.at

www.microtronics.at

Microtronics Engineering GmbH**(Headquarters)**

Hauptstrasse 7

3244 Ruprechtshofen

Austria, Europe

Tel. +43 (0)2756 77180

Fax. +43 (0)2756 7718033

office@microtronics.at

www.microtronics.at