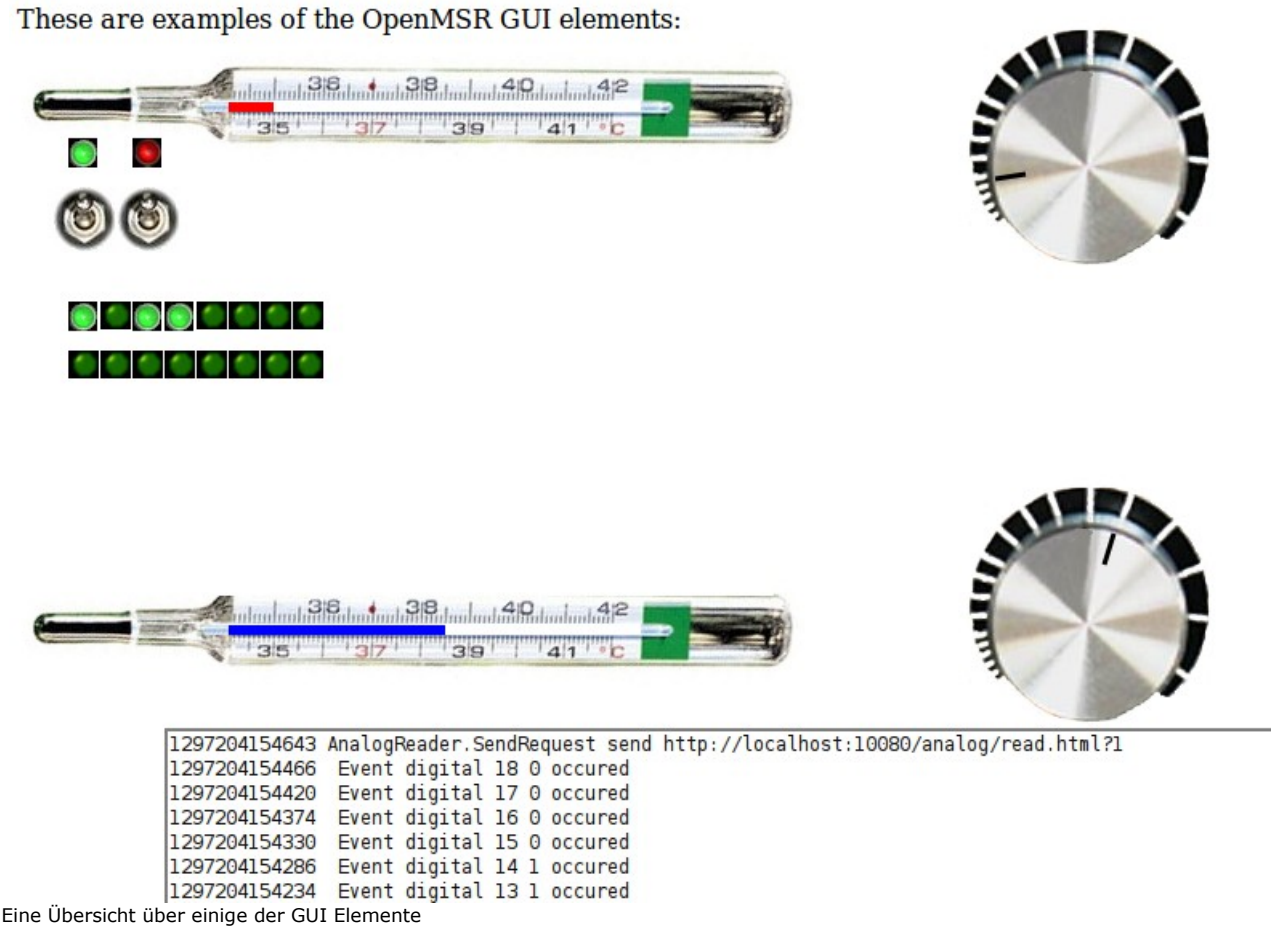
Die AJAX GUI Elemente



Eine Übersicht über einige der GUI Elemente

Zur einfachen und schnellen Gestaltungen von Bedienoberflächen und der Visualisierung von Steuerungen können sehr gut die GUI Elemente von OpenMSR genutzt werden.
Die Gui Elemente sind eine Sammlung von Javascript Elementen die mittels HTML5 und AJAX verbunden werden. Alle GUI Elemente sind dabei auf totale Flexibilität in ihrem Aussehen ausgerichtet. Durch einfachen Tausch der Elementgrafik kann aus dem silbernen Standarddrehknopf ein völlig anders aussehender Drehknopf werden. Aus einer LED kann ein Teil einer großen Grafik werden, der je nach Zustand verschiedene Grafiken zeigt. Damit aus diesen einzelnen Elementen eine funktionierende Oberfläche einer Steuerung wird bedarf es einer Technik die einzelnen Elemente miteinander zu verbinden. Dazu gehört auch die Kommunikation mit der Steuerung.

Im folgenden versuche ich die Grundlagen soweit zu erläutern, dass jedermann damit in die Lage versetzt wird eigene Oberflächen zu erstellen.

Übersicht über die möglichen Elemente einer OpenMSR GUI

Eingabeelemente – Schalter, Schieberegler, Drehknöpfe, Taster
Ausgabeelemente – LED, Glühbirne, Balkenanzeige, Zeigerinstrument, Füllstandsanzeige
Konnektoren – unsichtbare Elemente, die die GUI mit einer Steuerung verbinden.

Jedes dieser Elemente arbeitet dabei völlig autark und hat keine Informationen über andere Elemente. Damit daraus also eine funktionierende Oberfläche wird, braucht es einen internen

Kommunikationskanal: die Events. Ein Event ist einfach eine Nachricht, die aussagt, was sich ereignet hat. Das kann zum Beispiel „ Schalter 3 eingeschaltet" sein. Alle Eingabeelemente senden Events, alle Ausgabeelemente empfangen Events. Wenn also eine LED den Zustand eines Schalters anzeigen soll, so muss sie lediglich darauf eingestellt werden, auf den Event des Schalters zu reagieren. Bei jeder Änderung sendet der Schalter seinen Zustand als Event. Die LED empfängt diesen Event und ändert entsprechend ihren Zustand. Es darf zu jedem Event immer nur einen Sender geben, jedoch viele Empfänger. Die Konnektoren sind dabei spezielle Event-Empfänger bzw. Sender. Die Konnektoren empfangen entweder Events, die sie dann an die Steuerung weiterreichen, oder sie lesen Signale von der Steuerung und versenden sie als Events. Daher gibt es Konnektoren um digitale oder analoge Signale von der Steuerung zu lesen oder zu schreiben.
Wichtig ist dabei zu beachten, daß Events nicht an ein Ziel adressiert werden. Ein Event wird einfach gesendet – er hat kein spezielles Ziel.
Diese Technik macht die Erstellung von Oberflächen sehr einfach, obwohl Sie in der JavaScript Programmiersprache „programmieren" brauchen sie davon keine Ahnung zu haben. Sie erstellen lediglich eine HTML5 Seite, die Sie um die Aufrufe der GUI Elemente erweitern. Wie eine HTML Seite mittels CSS erstellt wird, müssen Sie jedoch beherrschen.

Alles was Sie dazu benötigen ist ein Texteditor. In diesem öffnen Sie jetzt die Seite „OpenMSR-GUI-Elements.html" im docroot Verzeichnis des DeviceServers. An Hand dieser Datei lernen wir das Erstellen von Oberflächen.
Der Anfang der Datei lautet:

```
<html>
  <head>
    <META HTTP-EQUIV="Pragma" CONTENT="no-cache">
    <META HTTP-EQUIV="Expires" CONTENT="-1">
    <title>OpenMSR GUI Elements</title>

   <style type="text/css">

      #TempMeterCanvas {
       background-color: white;
       position: absolute;
       left: 0px;
       top: 30px;
      }

      #knob1 {
       background-color: white;
       position: absolute;
       left  ….
```

Ein ganz normaler Header einer HTML Datei, was wichtig hier ist, sind die beiden META Tags. Der Browser soll möglichst nichts cachen. Was folgt sind die CSS Style Anweisungen für die einzelnen Elemente der GUI. Damit werden die Elemente auf der Oberfläche positioniert.

Später folgt:

```
    }
    #debug {
     background-color: white;
     position: absolute;
     left: 100px;
     top: 450px;
    }

</style>

<script type="text/javascript" src="js/openmsr.js"></script>

<script>
  function init() {

   OpenMSRInit();

   // show a customized horizontal meter as temp meter
   var MyTempMeter = new HorMeter('TempMeterCanvas','analog',1);
   MyTempMeter.width(525);
   MyTempMeter.height(84);
   MyTempMeter.canvbgimg('images/fieberthermometer.png');
   MyTempMeter.Resolution(10);
   MyTempMeter.maxVal(42.0);
   MyTempMeter.minVal(34.2);
   MyTempMeter.minPos(140,41);
   MyTempMeter.maxPos(380,41);
   MyTempMeter.color("red");
   MyTempMeter.MeterWidth(6);

   // a second meter
   var MyOtherMeter = new HorMeter('OtherMeter','analog',2);
   MyOtherMeter.width(525);
```

Nach den Styles folgt ein Script Aufruf, der die Bibliothek mit den Elementen lädt. In der Datei openmsr.js ist die komplette Logik aller Elemente enthalten.

Danach kommt ein zweiter Skript Aufruf, der die Elemente der GUI enthält. Alle Elemente der Seite werden dabei in der Funktion „init“ aufgeführt. Die Funktion muss jedoch zuerst die OpenMSR Elemente starten, was durch den Befehl „OpenMSRInit();“ geschieht.

Mittels „`var MyTempMeter = new HorMeter('TempMeterCanvas','analog',1);`“ wird ein neues Fieberthermometer erzeugt, das auf der Fläche TempMeterCanvas angezeigt wird und auf den analogen Event 1 reagieren soll. Die weiteren Zeilen konfigurieren dieses Fieberthermometer und weisen ihm verschiedene Eigenschaften zu. Zu jedem Element finden sie die genauen Parameter im Referenzkapitel.