

⟨R⟩ - Control Plus IP 8

Stand: 24.03.2020

Gibt die MAC-Adresse des Gerätes als Rückgabewert aus.

GET:

- <https://192.168.0.6/cgi-bin/api/v1/discover>

Antwort:

```
{
  „mac“: „C4:93:00:0A:EE:0C“
  „name“: „Rutenbeck R-Control IP 8“
}
```

Generiert einen neuen Sicherheitsschlüssel und gibt diesen aus. Damit das Gerät auf diese Anfrage reagiert muss zunächst unter System → Administration, unter „Freischalten des REST API“ der Button „Freischalten“ betätigt werden oder der Resettaster an dem Gerät selbst wird für 1-5 Sekunden betätigt (Die WLAN LED und LAN LED blinken im Sekunden Takt abwechselnd solange der Sicherheitsschlüssel abgefragt werden kann). Nach Speichern und Anwenden kann 60 Sekunden lang ein neuer Sicherheitsschlüssel generiert und ausgegeben werden.

POST:

- <https://192.168.0.6/cgi-bin/api/v1>

Body:

```
{
  „devicetype“:“my application“
}
```

Antwort:

```
[
  {
    „success“: {
      „username“: „ed21dg54d238wer21cs654rtb7612adf“
    }
  }
]
```

Ist ein Sicherheitsschlüssel bereits generiert, kann der generierte Sicherheitsschlüssel abgefragt werden. Somit kann ein neuer Nutzer hinzugefügt werden. Die Schnittstelle muss, wie bei der Generierung eines neuen Sicherheitsschlüssels, freigegeben werden. Über die Weboberfläche oder über den Resettaster am Gerät.

GET:

- <https://192.168.0.6/cgi-bin/api/v1/apikey>

Antwort:

```
[
  {
    „success“: {
      „username“: „ed21dg54d238wer21cs654rtb7612adf“
    }
  }
]
```

Nach der Generierung oder Abfrage des Sicherheitsschlüssels muss dieser als Authorization Header bei jeder Anfrage an das Gerät mit gesendet werden.

Authorization: ed21dg54d238wer21cs654rtb7612adf

Startet die Callback Funktion, bei der Zustandsänderungen der Eingänge, Ausgänge und der Anloeingänge im JSON Format an die übermittelte URL gesendet werden. Es ist wichtig, dass http oder https in der URL enthalten sind. Bei Start der Callback Funktion werden einmal alle Zustände übermittelt.

POST:

- <https://192.168.0.6/cgi-bin/api/v1/callback>

Body:

```
{
  „url“: „http://send/data/to/this/resource“
}
```

Antwort:

```
{
  "success": {
    "/callback": "Started on https://192.168.0.6/callbacktesting"
  }
}
```

Callback Funktion beenden und übermittelte URL löschen.

GET:

- <https://192.168.0.6/cgi-bin/api/v1/callbackdel>

Antwort:

```
{
  "callback": "deleted"
}
```

Die Nachrichten der Callback Funktion enthält den Header Content-Type: application/json und Content-Length: 23 mit der Länge der Nachricht

Antwort:

```
{
  "1": {
    "output": 2,
    "value": false
  },
  "2": {
    "output": 3,
    "value": true
  },
  "3": {
    "input": 1,
    "value": false
  },
  "4": {
    "input": 2,
    "value": true
  },
  "5": {
    "analog": 2,
    "value": 785
  },
  "6": {
    "analog": 1,
    "value": 1961
  }
}
```

Alle Ausgänge auslesen

GET:

- <https://192.168.0.6/cgi-bin/api/v1/outputs>

Antwort:

```
{
  „1“: {
    „state“: {
      „value“: 0
    },
    „name“: „Ausgang 1“
  },
  „2“: {
    „state“: {
      „value“: 0
    },
    „name“: „Ausgang 2“
  },
  .
  .
  .
  „7“: {
    „state“: {
      „value“: 0
    },
    „name“: „Ausgang 7“
  },
  „8“: {
    „state“: {
      „value“: 0
    },
    „name“: „Ausgang 8“
  }
}
```

Einzelne Ausgänge auslesen

GET:

- <https://192.168.0.6/cgi-bin/api/v1/outputs/1>

Antwort:

```
{
  „state“: 0,
  „name“: „Output 1“
}
```

Alle Eingänge auslesen

GET:

- <https://192.168.0.6/cgi-bin/api/v1/inputs>

Antwort:

```
{
  „1“: {
    „state“: {
      „value“: 0
    },
    „name“: „Eingang 1“
  },
  „2“: {
    „state“: {
      „value“: 0
    },
    „name“: „Eingang 2 “
  },
  .
  .
  .
  .
  .
  .
  .
  .
  .
  .
  .
  „15“: {
    „state“: {
      „value“: 0
    },
    „name“: „Eingang 15“
  },
  „16“: {
    „state“: {
      „value“: 0
    },
    „name“: „Eingang 16 “
  }
}
```

Einzelne Eingänge auslesen

GET:

- <https://192.168.0.6/cgi-bin/api/v1/inputs/1>

Antwort:

```
{
  „state“: 0,
  „name“: „Input 1“
}
```

Einzelne Ausgänge steuern

POST:

- <https://192.168.0.6/cgi-bin/api/v1/outputs/1/state>

Body:

```
{
  „value“:1
}
```

Antwort:

```
[
  {
    „success“: {
      „/outputs/1/state“: 1
    }
  }
]
```

Alle Analogeingänge auslesen. Als Antwort kommt der Digitalwert des Analog Digital Konverters.

GET:

- <https://192.168.0.6/cgi-bin/api/v1/analoginputs>

Antwort:

```
{
  „1“: {
    „state“: {
      „value“: 0
    },
    „name“: „Analogeingang 1“
  },
  „2“: {
    „state“: {
      „value“: 0
    },
    „name“: „Analogeingang 2“
  }
}
```

Einzelne Analogeingänge auslesen

GET:

- <https://192.168.0.6/cgi-bin/api/v1/analoginputs/1>

Antwort:

```
{
  „state“: {
    „value“: 0
  },
  „name“: „Analogeingang 1“
}
```

Alle Analogeingänge auslesen. Als Antwort kommt der aktuelle Spannungswert.

GET:

- <https://192.168.0.6/cgi-bin/api/v1/analoginputsvolt>

Antwort:

```
{
  „1“: {
    „state“: {
      „value“: 5.123541298451235
    },
    „name“: „Analogeingang 1“
  },
  „2“: {
    „state“: {
      „value“: 5.123541298451235
    },
    „name“: „Analogeingang 2“
  }
}
```

Einzelnen Spannungswert eines Analogeingangs auslesen

GET:

- <https://192.168.0.6/cgi-bin/api/v1/analoginputsvolt/1>

Antwort:

```
{
  „state“: {
    „value“: 5.123541298451235
  },
  „name“: „Analogeingang 1“
}
```


Alle Analogeingänge auslesen. Als Antwort kommt der Vorzeichenbehaftet wert, der Skala, die der Nutzer in der Weboberfläche konfiguriert hat. Außerdem die vom Nutzer konfigurierte Einheit. Die werte sind auf zwei Nachkommastellen beschränkt.

GET:

- <https://192.168.0.6/cgi-bin/api/v1/analoginputsunit>

Antwort:

```
{
  „1“: {
    „state“: {
      „value“: -10.5
    },
    „name“: „Analogeingang 1“,
    „unit“: „°C“
  },
  „2“: {
    „state“: {
      „value“: 5.12
    },
    „name“: „Analogeingang 2“,
    „unit“: „V“
  }
}
```

Einzelnen Spannungswert eines Analogeingangs auslesen

GET:

- <https://192.168.0.6/cgi-bin/api/v1/analoginputsunit/1>

Antwort:

```
{
  „state“: {
    „value“: 5.123541298451235
  },
  „name“: „Analogeingang 1“
}
```

Ausführen konfigurierter Verbindungen. Führt einen kurzen virtuellen Tastbefehl aus. Durch diesen kurzen Tastbefehl wird die die Konfiguration einer Verbindung Schritt für Schritt ausgeführt.

POST:

- <https://192.168.0.6/cgi-bin/api/v1/virtualoutputs/1/state>

Body:

```
{
  „value“:1
}
```

Antwort:

```
[
  {
    „success“: {
      „/virtualoutputs/1/state“: 1
    }
  }
]
```

Abfragen nach konfigurierter Verbindungen

GET:

- <https://192.168.0.6/cgi-bin/api/v1/connections>

Antwort:

```
{
  "1": 1,
  "2": 1,
  "3": 1,
  "4": 0,
  "5": 0,
  "6": 0,
  "7": 0,
  "8": 0
}
```