eGauge XML API

(v1.51)

eGauge Systems LLC

May 3, 2017

1 Overview

이 문서는 CGI 쿼리를 사용하여 eGauge 장치에서 원시 XML 데이터를 읽는 방법을 설명합니다. 이 문서는 펌웨어 버전 v1.00 이상에 적용됩니다. 새 쿼리 매개 변수 또는 푸시 옵션에 대한 지원을 얻으려면 펌웨어 업그레이드가 필요할 수 있습니다.

두 가지 유형의 쿼리가 있습니다 : 즉각적인 쿼리와 저장된 데이터 쿼리. 전자는 모든 측정 된 데이터의 가장 최근 값을 읽는 반면 후자는 eGauge 장치에 내장 된 데이터베이스에 저장된 히스토리 데이터의 일부를 읽습니다.

숫자 데이터는 정수 문자열 또는 부동 소수점 문자열로 반환됩니다. 정수 문자열의 기본 형식은 부호없는 32 비트 정수 (U32) 또는 부호있는 64 비트 정수 (S64)입니다. U32의 범위는 0에서 4,294,967,295까지입니다. S64의 범위는 -9,223,372,036,854,775,808에서 9,223,372,036,854,775,807까지입니다. 부동 소수점 문자열의 기본 형식은 IEEE-754 64 비트 부동 소수점 형식입니다.

S64 값은 원형입니다. 최대 양수 값에 도달하면 가장 작은 음수 값으로 랩 어라운드합니다 (반대의 경우도 마찬가지입니다). 자바 스크립트는 기본적으로 64 비트 값을 처리 할 수 없으며 오버플로를 방지하기 위해주의를 기울여야합니다.

Unix 타임 스탬프는 U32 형식을 사용하고 정수 레지스터 값은 S64 형식을 사용합니다. 10 진수 값은 IEEE-754 64 비트 부동 소수점 형식을 사용합니다.

2 Instantaneous Data

순시 데이터는 1 초에 한 번 업데이트됩니다. URI 참조를 통해 반입됩니다.

```
/cgi-bin/egauge?params
```

매개 변수에 사용할 수있는 값은 다음 절에서 설명합니다. 곱하기 쿼리 매개 변수는 앰퍼샌드로 구분하여 지정할 수 있습니다 (예 : inst 및 tot으로 inst와 tot을 모두 지정).

2.1 Query Parameters (instantaneous data)

Parameter:	Type:	Description:
tot	n/a	실제 등록에서 계산 된 합계 및 가상 레지스터가 출력에 포함됩니다.
noteam	n/a	로컬 측정 값만보고하도록 요청합니다 (레거시 형식). 그만큼 원격 장치 및 비 전력 레지스터에서 얻은 모든 레지스터의 값은 생략됩니다. 사용을 권장하지 않습니다.
teamstat	n/a	팀 구성 상태가보고되도록 요청합니다. 여기에는 "초과"값 등록이 포함됩니다.
v1	n/a	레거시 v0.01 형식이 아닌 v1.00 형식으로 출력되도록 요청합니다. 펌웨어 v1.2 이상은 기본적으로 v1.00 출력으로 설정됩니다.
inst	n/a	정상 레지스터 값과 함께 각 레지스터의 순간 변화율이보고되도록 요청합니다.

2.2 Instantaneous Data (v1.00 format)

v1.00 형식의 출력 예가 그림 1에 나와 있습니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<data serial="0x78666e4d">
 <ts>1284607004</ts>
 <r t="P" n="Grid"><v>5196771697</v></r>
 <r t="P" n="Solar"><v>21308130148</v></r>
 <r t="P" n="Grg&amp;amp;Bth (PHEV)"><v>17601054087</v></r>
</data>
```

Figure 1: Example of instantaneous data with **v1** query parameter.

순간 데이터 쿼리는 데이터 시작 및 종료 태그로 묶인 단일 요소를 반환합니다. 데이터 요소는 구성 일련 번호를 16 진수 문자열로 지정하는 serial 속성을 가질 수 있습니다. 이 일련 번호는 장치 구성이 변경 될 때마다 변경됩니다. 따라서 일련 번호를 사용하여 구성 변경 사항을 감지 할 수 있습니다.
이 구성 일련 번호는 장치 하드웨어 일련 번호와 아무 관련이 없으며 장치의 고유 한 식별자가 아닙니다.

데이터 요소 내에서 다음 요소가 나타날 수 있습니다.

Element Name:	Type:	Description:
ts	Integer (U32)	보고 된 측정 값을 얻은 장치 - 로컬 시간. UNIX 시간 소인입니다 (1970 년 1 월 1 일 UTC 시작 이후 초).
r	Struct	장치에 구성된 레지스터 당 하나의 r 요소가 있습니다. 속성 t는 레지스터의 타입을 식별하는 코드를 지정한다 (2.2.1 절 참조). Attribute n은 레지스터 이름을 지정하며 특수 문자를 인코딩하는 HTML 엔티티가 포함될 수 있습니다. 속성 rt는 합계 문자열로 설정되어 레지스터가 물리적 레지스터에서 값이 계산 된 총 또는 가상 레지스터임을 나타낼 수 있습니다. 각 요소에 대해 두 개의 하위 요소가 나타날 수 있습니다 (v 및 i).
v	Integer (S64)	유형별 단위로 표현 된 누적 레지스터 값. 두 개의 연속 판독 값을 빼고 샘플 사이의 경과 시간 (초)으로 나눈 값은 레지스터에 대한 평균 변화율을 제공합니다.
i	Float	가장 최근의 1 초 간격으로 측정 된 레지스터 값의 평균 변화율.

2.2.1 Register Types

아래 표는 레지스터 유형, 코드로 표시되는 물리량 및 레지스터 값 변경 비율에 대한 측정 단위를 식별하는 데 사용되는 코드를 지정합니다.

Code:	Physical Quantity:	Rate-of-change Unit:	Recording Unit:
Ee	Irradiance	W/m^2 (Watts per square meter)	W/m^2
F	Frequency	Hz (Hertz)	mHz
I	Current	A (Ampère)	mA
PQ	Reactive Power	*var* (Volt-Ampère reactive)	*var*
Pa	Pressure	*Pa* (Pascal)	*Pa*
P	Power	W (Watt)	W
Qv	Volumetric flow	mm^3/s	mm^3/s
Q	Mass-flow	g/s (gram per second)	g/s
R	Resistance	Ω (Ohm)	Ω
S	Apparent Power	VA (Volt-Ampère)	VA
THD	Total Harmonic Distortion	%	m%
T	Temperature	C (Centigrade Celsius)	mC
V	Voltage	V (Volt)	mV
#	Numeric	(unit-less)	(unit-less)
$	Monetary	*currency unit/s*	$2^{-29} \cdot$ *currency unit/s*
a	Angle	° (degrees)	$m°$
h	Relative humidity	%	0.1%
v	Speed	m/s (meter per second)	mm/s
Qe	Charge	Ah (Ampère-hours)	mAh

시간이 지남에 따라 새로운 레지스터 유형 코드가 추가 될 수 있습니다. eGauge XML 데이터를 처리하는 소프트웨어는 알 수없는 레지스터 유형 코드가 발생할 때 정상적으로 저하되도록 작성되어야합니다.

위의 단위는 레지스터의 변경 비율에 적용됩니다. 레지스터의 값은 변화율에 대한 시간 - 적분이므로 레지스터의 단위는 위의 단위에 시간 (초)을 곱한 값입니다. 예를 들어, 전력의 경우, 변화율 단위는 와트이므로 레지스터 값은 와트 초 (줄과 동일)입니다. 와트 초는 3,600,000으로 나눠 킬로와트시 (kWh)로 변환 할 수 있습니다.

데이터베이스는 "레코딩 유닛"열에 표시된 입도로 부호있는 64 비트 수량으로 모든 데이터를 기록합니다. 이에 따라 저장된 데이터 (3 절 참조)는 이러한 정수 값을보고하고보다 사용자 친화적 인 변화율 단위를 얻기 위해 적절한 크기 조정을 적용해야 할 수 있습니다.

2.3 Team Status

격 장치를 읽도록 구성된 eGauge 장치는 구성원이 장치 자체 및 모든 원격 장치를 포함하는 팀의 일부입니다.

이러한 팀의 상태는 teamstat 쿼리 매개 변수를 전달하여 얻을 수 있습니다. 반환 된 상태는 구성된 레지스터의 가용성 및 상태를 나타내며 그 중 일부는 하나 이상의 원격 장치에서 가져올 수 있습니다. 팀 상태 형식의 출력 예가 그림 2에 나와 있습니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<status>
 <lag unit="ms">227</lag>
 <reg>
  <name>Grid</name>
  <available>1</available>
  <last_update>1312472842</last_update>
  <excess>0</excess>
  <last_val>0</last_val>
 </reg>
   :
 <reg>
  <name>Solar</name>
  <available>0</available>
  <last_update>1312472842</last_update>
  <excess>0</excess>
  <last_val>0</last_val>
 </reg>
</status>
```

Figure 2: Example of team status data (**teamstat** query parameter).

팀 상태는 상태 시작 및 종료 태그로 묶인 단일 요소로 반환됩니다. status 요소 내에 다음 요소가 나타날 수 있습니다.

Element Name:	Type:	Description:
lag	Integer (U32)	보고 된 순간 데이터가 실시간보다 뒤처지는 시간입니다. 이 지연은 일반적으로 밀리 초 단위로보고되며 특성 단위에 대한 ms 값으로 표시됩니다. 이상적으로이 지체는 0에 가까워 야하지만 도달하기가 느리거나 일시적으로 사용할 수없는 원격 장치에서 데이터를 가져 오는 경우 더 클 수 있습니다
reg	Struct	디바이스에 대해 정의 된 각 레지스터에 대해 이러한 요소가 하나 있습니다. 이러한 엘레멘트는 v1.00 인스턴트 데이터 형식의 레지스터 (r) 태그와 동일한 순서로 나타납니다. 하위 요소 이름, 사용 가능, 마지막 업데이트, 초과 및 마지막 val은이 요소 내부에 나타날 수 있습니다.
available	Boolean	이 레지스터에 대한 데이터를 제공하는 (원격) 장치가 현재 도달 가능한지 여부를 나타냅니다. 값 1은 장치에 도달 할 수 있음을 나타내고 값 0은 장치에 도달 할 수 없음을 나타냅니다.
last update	Integer (U32)	이 레지스터 값이 마지막으로 갱신 된 UNIX 시간 소인.
excess	Integer (S64)	이 요소에 0이 아닌 값은이 레지스터에 대한 데이터를 제공하는 장치가 얼마 전에 도달 할 수 없었고 현재 레지스터 값이 실제 값에서 벗어나는 양을 나타냅니다.
last_val	Integer (S64)	이 레지스터에 기록 된 마지막 값

3 Stored Data

저장된 데이터는 1 분에 1 번 업데이트됩니다. URI 참조를 통해 반입됩니다.

```
/cgi-bin/egauge-show?params
```

이 쿼리는 에너지 데이터를 열의 행으로 반환합니다. 각 행은 특정 시점에 대한 데이터를보고합니다. 행은 고정 된 수의 열과 구성된 레지스터 당 하나의 열로 구성됩니다. 다양한 쿼리 매개 변수 params를 지정하여 검색 할 데이터와 반환 할 형식을 선택할 수 있습니다.

3.1 Query Parameters (stored data)

Parameter:	Type:	Description:
a	n/a	물리적 레지스터 값에서 계산 된 합계 및 가상 레지스터가 각 행의 첫 번째 열로 포함되도록 요청합니다. 이 값은 장치에 대해 구성된 총계 및 가상 레지스터 규칙에 따라 계산됩니다.
E	n/a	에포크를 기준으로 값이 출력되도록 요청합니다. 즉, "녹화가 시작된 날짜와 시간"의 값은 0이됩니다.
b	n/a	출력을 데이터 백업 형식으로 반환하도록 요청합니다.
c	n/a	출력을 CSV (쉼표로 구분 된 값) 형식으로 반환하도록 요청합니다.
e	n/a	요청 된 범위를 벗어나는 하나의 추가 데이터 포인트 출력을 요청합니다. 이것은 유사하지만 동일하지는 않지만 매개 변수 n에 대해 N + 1 값을 전달하는 것과 유사합니다. 두 가지가 동일하지 않은 이유는 데이터베이스 기반 세밀도가 요청한 것보다 더 거칠 수 있기 때문입니다. 예를 들어 데이터를 1 분 단위로 요청할 수 있지만 데이터베이스에 1 시간 단위 만 사용할 수있는 경우 매개 변수 n에 N + 1을 전달하면 아무 효과가 없지만 e는 시간별 데이터 포인트가 마지막으로 요청 된 데이터 포인트도 출력에 포함됩니다. 이 매개 변수는 T 또는 w 매개 변수가 지정 될 때 적용되지 않습니다. 이 매개 변수는 펌웨어 v1.2에서 도입되었습니다.
m	n/a	n 및 s 매개 변수가 분 단위로 지정되도록 지정합니다.
h	n/a	n 및 s 매개 변수가 시간 단위로 지정되도록 지정합니다.
d	n/a	n 및 s 매개 변수가 일 단위로 지정되도록 지정합니다.
S	n/a	n 및 s 매개 변수가 초 단위로 지정되도록 지정합니다.
C	n/a	반환 된 데이터가 델타 압축되도록 지정합니다. 즉, 데이터의 첫 번째 행 이후에, 각 후속 행의 열은 이전 행의 열 값과 관련하여 차이로 표현됩니다. CSV (c) 매개 변수와 결합하면 첫 번째 행이 항상 생략됩니다.
n	Integer (U32)	리턴되는 최대 행 수를 지정합니다.
s	Integer (U32)	행을 출력 한 후 건너 뛸 행 수를 지정합니다. 예를 들어 h & s = 23은 행이 출력 된 후 23 시간 분량의 데이터를 건너 뛰고 d와 같습니다.
f	Integer (U32)	반환 될 첫 번째 행의 타임 스탬프를 지정합니다.
t	Integer (U32)	리턴 될 마지막 행의 시간 소인을 지정합니다.
w	Integer (U32)	지정된 타임 스탬프보다 새로운 데이터 만 반환하도록 요청합니다. 시간 소인이 미래에있을 경우, u 리는 즉시 완료되어 대기 시간 속성이 지정된 시간 소인보다 작은 데이터가 사용 가능하게되기까지 경과되어야하는 시간 (초)을 표시하는 빈 데이터 요소를 리턴합니다.
T	Integer-list (U32)	데이터 행을 반환 할 감소 값 (더 오래된 것부터 이전 값)으로 정렬 된 시간 소인 목록을 지정합니다.
Z	string	CSV 데이터를 내보낼 때 사용할 표준 시간대를 지정합니다. 이 문자열의 형식은 환경 변수 TZ에서 http://www.opengroup.org/onlinepubs/ 009695399 / basedefs / xbd chap08.html에 설명되어 있습니다. 펌웨어 v1.12부터는이 매개 변수의 값을 생략 할 수 있습니다. 이 경우 장치는 장치 로컬 시간대 ("날짜 및 시간"대화 상자에서 "시간대"설정을 통해 지정)를 사용하여 타임 스탬프를 변환합니다.

3.2 Returned XML Data

매개 변수 m & n = 3을 사용하는이 쿼리의 출력 예가 그림 3에 나와 있습니다.

저장된 데이터 쿼리는 그룹 시작 및 종료 태그로 묶인 단일 요소를 반환합니다. 순시 응답의 데이터 요소와 마찬가지로 그룹 요소에는 구성 일련 번호를 나타내는 일련의 속성이있을 수 있습니다.

이 구성 일련 번호는 장치 하드웨어 일련 번호와 아무 관련이 없으며 장치의 고유 한 식별자가 아닙니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<group serial="0x37cdd096">
<data columns="3" time_stamp="0x4c9197e4" time_delta="60" epoch="0x47395980">
 <cname t="P">Grid</cname>
 <cname t="P">Solar</cname>
 <cname t="P">Grg&amp;amp;Bth (PHEV)</cname>
 <r><c>5203642184</c><c>21308125431</c><c>17598056700</c></r>
 <r><c>5203503484</c><c>21308125526</c><c>17598116405</c></r>
 <r><c>5203368999</c><c>21308125626</c><c>17598176060</c></r>
</data>
</group>
```

Figure 3: Example of stored data.

가장 최근의 값은 데이터 요소에서 먼저보고되고 다음 행은 이전 행보다 시간 △ 초가됩니다.

group 요소 내에서 다음 요소가 나타날 수 있습니다.

Element Name:	Type:	Description:
data	Struct	이러한 요소 중 하나는 데이터 행의 각 연속 시퀀스에 대해 나타납니다. At-tribute 열은 각 행의 열 수를 지정합니다. 속성 시간 소인은 첫 x 째 행에 대한 UNIX 시간 소인 (16 진수)을 지정합니다. 속성 시간 델타는 다음 행의 시간 소인을 얻기 위해 빼기 시간 (초)을 지정합니다. 속성 신기원은 기록이 시작된 시간의 UNIX 시간 소인 (16 진수)을 지정합니다. 데이터 행이 델타 인코딩 된 경우 속성 델타는 true와 같습니다 (아래 참조). 속성 대기 시간은 w 매개 변수로 지정된 시간 소인이 읽기 위해 사용되기까지 경과해야하는 시간 (초)을 지정합니다.
cname	String	증가하는 열 순서로 열의 레지스터 이름을 지정합니다. 이 엘레멘트는 첫 번째 데이터 요소에만 나타날 수 있습니다. 후속 데이터 요소에서 레지스터 이름은 첫 번째 것과 동일하게 유지되어야합니다. 이 요소는 레지스터의 유형을 식별하는 t 속성을 가질 수있다 (2.2.1 참조). t 속성이 없으면 P (전원)의 유형 코드를 가정해야합니다.
r	Struct	하나의 데이터 행을 포함합니다.
c	Integer (S64)	개별 누적 레지스터 값입니다. 이 값은 해당하는 cname 선언에 의해 지정되거나 암시 된 레지스터 유형에 따라 해석되어야합니다

4 Push Data Setup

전 섹션에서 설명한 것처럼 eGauge 장치에서 데이터를 폴링하는 것 외에도 eGauge를 설정하여 데이터를 임의의 URI로 푸시 (push) 할 수 있습니다. http (암호화되지 않음) 및 https (암호화 된) 구성표가 모두 지원됩니다. 데이터는 섹션 3에서 설명한 것과 동일한 XML 형식으로 푸시되고 HTTP POST의 본문으로 전송됩니다 (폼이 아님). 이 POST의 Content-Type은 text / xml입니다.

URI는 데이터를 푸시해야하는 웹 서버를 지정합니다. 지정된 방식에 관계없이이 전송은 HTTP / 1.1 청크 분할 전송 인코딩 (즉, Content-Length 헤더 없음)을 사용합니다. 선택적으로, 데이터는 content-encoding deflate 또는 gzip으로 압축 될 수 있습니다 (아래의 push-options 참조). 콘텐트 인코딩 (즉, 압축)은 전송 인코딩 전에 적용된다는 점에 유의해야한다. 일반적으로 웹 서버는 전송 인코딩을 디코딩하므로 최종 수신자는 콘텐츠 인코딩 (있는 경우) 만 처리하면됩니다.

웹 서버에 데이터를 푸시하기 위해 eGauge를 설정하는 방법에는 수동 (사용자 정의) 또는 푸시 서비스 정의를 통한 두 가지가 있습니다. 수동 설정은 URI, 푸시 업데이트 사이의 시간 간격 및 데이터를 푸시해야하는 옵션 (예 : 가상 레지스터를 푸시해야하는지 여부를 지정)을 포함하여 모든 통신 매개 변수를 지정하는 것을 포함합니다. 수동 설치는 최종 사용자에게는 번거롭고 오류가 발생하기 쉽습니다. 푸시 서비스 정의를 사용하는 것이 훨씬 사용자 친화적이며 타사 서비스 제공 업체가 서버 측에서 eGauge의 데이터를 수락하는 데 필요한 로그인 또는 로그인 절차를 통해 고객을 자동으로 안내할 수 있습니다.

푸시 서비스는 설정 일반 설정의 데이터 공유에서 장치에 구성됩니다. 자동 푸시 서비스 정의가 사용되면 eGauge가 푸시하는 URI가 표시되고 옵션이 숨겨집니다.

푸시 간격 (푸시 업데이트 간격)은 푸시되는 데이터의 세분성보다 짧을 수 있습니다. 푸시 간격에 새로운 데이터가 사용 가능한 경우에만 푸시 URI에 대한 연결이 설정됩니다. 이는 신뢰할 수없는 링크에서의 데이터 전송 안정성을 향상시키는 데 사용될 수 있습니다. 예를 들어 푸시 간격이 1 분이고 데이터 단위가 1 시간 인 경우 기기는 데이터가 업로드 될 때까지 분당 한 번씩 새 데이터를 푸시하고 다음 시간별 데이터 포인트가 연결되기 전에 사용할 수있게 될 때까지 기다립니다 푸시 URI.

달리 지정하지 않는 한, 데이터는 분 단위로 누를 수 있습니다 (사용 가능한 정도까지).

4.1 Automatic Push Data Setup via Push Service Definition

푸시 서비스 정의를 설정하려면 타사 서비스 공급자가 eGauge Systems LLC에 다음 정보를 제공해야합니다.

• 서비스 이름 (임의의 텍스트 문자열이지만 길이는 24 자 이하 여야 함).
• 타사 서비스로 장치를 등록하는 데 사용할 제어 URI (cURI).

이 정보가 eGauge Systems LLC에 의해 수신되고 처리되면 사용자는 브라우저에서 장치의 웹 인터페이스를 열고 설정 일반 설정을 클릭하여 타사 서비스로 장치에 등록 할 수 있습니다. 데이터 공유 아래의 드롭 다운 목록에서 타사 서비스를 선택할 수 있습니다. 사용자가 [저장] 버튼을 클릭하면 선택한 푸시 서비스가 변경되면 다음 작업이 수행됩니다.

1. eGauge는 선택된 푸시 서비스의 이름을 저장합니다. 이 단계를 완료하기 전에 적절한 자격 증명을 사용하여 사용자를 인증해야 할 수 있습니다 (예 : 사용자 이름 "소유자"및 암호 "기본값", 장치에 대한 직접 / LAN 연결이 있다고 가정). eGauge는 다음 단계가 성공적으로 완료 될 때까지 푸시 서비스를 비활성으로 표시합니다.

2. eGauge는 사용자의 브라우저를 서비스 제공 업체의 cURI로 리디렉션하여 요청에 다양한 GET 매개 변수를 전달합니다. 특히 브라우저는 URI :

CURI? 활성화 & mfg = eGauge & did = 이름 & sn = SN & regs = rlist & virt = vlist & ruri = rURI & opts = opts

어디에:
cURI : 타사 서비스 공급자가 지정한 제어 URI입니다.

name : 장치의 호스트 이름 (device-id)입니다. 호스트 이름은 최종 고객이 변경할 수 있습니다. 호스트 이름은 장치가 프록시 서버에 연결되어 있고 해당 프록시 서버에 대해서만 고유 한 것이 확실한 경우에만 고유해야합니다.

SN : 고유 번호가 보장되고 최종 고객이 변경할 수없는 장치의 일련 번호입니다. 그러나 일련 번호는 예를 들어 장치에 장애가 발생하여 교체 장치로 교체 된 경우와 같이 변경 될 수 있습니다. 이 매개 변수는 최신 장치 (eGauge3 시리즈 이상) 및 펌웨어 v2.1 이상을 사용하는 경우에만 포함됩니다.

rlist : 장치가 기록하고있는 레지스터 이름의 쉼표로 구분 된 목록입니다. 각 이름은 공백이 더하기 기호 (+)로 바뀌고 백분율 기호 (%)와 그 뒤에 바이트 값을 인코딩하는 16 진수 문자열로 다른 비영 문자 바이트로 대체되는 URI 인코딩입니다.

vlist : 장치에 대해 정의 된 가상 레지스터 이름의 쉼표로 구분 된 목록입니다. 각 이름은 URI로 인코딩됩니다.

rURI : 반환 URI : 타사 서비스를 사용하여 가입 프로세스가 성공적으로 완료되면 브라우저에서 지시 할 URI입니다. 이 값은 URI로 인코딩됩니다. 현재이 값은
http : // DEV-URI / cgi-bin / protected / egauge-cfg? setpush 여기서 DEV-URI는 푸시를 구성하는 클라이언트가 사용하는 URI이며 로컬 IP 주소 또는 완전한 프록시 서버 URI 일 수 있습니다. 이는 변경 될 수 있으며 여러 기기에서 동일하다고 가정해서는 안됩니다.

opts : 장치 펌웨어가 지원하는 모든 푸시 옵션의 쉼표로 구분 된 목록입니다.

3. 웹 브라우저가 cURI를 열면 타사 서비스는 장치에서 데이터 수신을 준비하는 데 필요한 모든 단계를 수행 할 수

있습니다 (예 : 사용자 / 장치에 대한 새 계정 만들기 또는 기존 계정을 장치와 연결) . 성공적으로 완료되면 타사 서비스는 인코딩 유형 multipart / form-data를 사용하여 HTTP POST를 통해 브라우저를 rURI로 다시 리디렉션해야합니다. 게시 된 양식은 다음 부품 이름에 대한 값을 전달할 수 있습니다.

• uri : eGauge가 데이터를 푸시해야하는 URI입니다. 예 :
https://datacenter.example.com/push-data?01345a535a
• 간격 : 업데이트 간격 (초).

옵션 : 데이터를 푸시해야하는 옵션입니다. 쉼표로 구분하여 여러 옵션을 지정할 수 있습니다. 사용 가능한 옵션은 다음과 같습니다.

- totals : 또한 가상 (계산 된) 레지스터의 값을 푸시합니다.
- 초 : 초 단위로 데이터를 누릅니다.
- 시간 : 시간 단위로 데이터를 밀어 넣습니다 (가능한 정도까지).
- day : 하루 단위로 데이터를 밀어 넣습니다 (가능한 정도까지).
- 최대 = N : N 개의 데이터 행을 밀어 넣습니다. N은 900을 초과 할 수 없습니다.
- 건너 뛰기 = N : 데이터 행을 누른 후에 다른 N 행을 건너 뛰기 전에 건너 뜁니다. 예를 들어 미세한 입도로 "skip = 14"는 15 분마다 하나의 데이터 행을 푸시하게됩니다.
- secure : https 스키마를 사용할 때 인증서 확인 오류는 대개 무시됩니다. 이 옵션을 지정할 때 웹 서버 인증서가 올바르게 확인되어야하며, 그렇지 않으면 데이터가 푸시되지 않습니다.
- deflate : 압축 알고리즘을 사용하여 푸시 된 데이터를 압축합니다.
- gzip : gzip 알고리즘을 사용하여 푸시 된 데이터를 압축합니다.
- old first : 새 데이터보다 오래된 데이터를 보류합니다. 일반적으로 가장 최근 N 개의 데이터 행이 푸시됩니다 (max = N 옵션에 지정된대로). 그러나 이전 첫 번째 옵션이 지정되고 아직 푸시되지 않은 N 개 이상의 행 데이터가있는 경우 이전 데이터가 먼저 전송됩니다. 데이터를 푸시 할 수있는 것보다 빠르게 생성하는 경우이 옵션을 사용하면 푸시 된 데이터가 현재 시간보다 멀리 떨어져 더 멀리 떨어지는 효과를 얻을 수 있습니다.
- epoch : 에포크를 기준으로 값을 밀어 넣습니다 (egauge-show 매개 변수 "E"참조).

참고 : deflate 및 gzip 옵션은 펌웨어 버전 2.03부터 지원됩니다. 오래된 펌웨어는 이러한 옵션을 무시하고 압축되지 않은 데이터를 보냅니다.

user : HTTP 기본 인증을 통한 푸시 중에 제공 될 사용자 이름입니다. 사용자 이름을 지정하지 않으면 기본 인증 정보가 제공되지 않습니다.

• pw : HTTP 기본 인증을 통한 푸시 중에 제공 할 암호입니다.

2. eGauge는 POST 된 양식에서 매개 변수를 추출하여 저장합니다. 모든 매개 변수에 유효한 값이 있다고 가정하면 eGauge는 타사 서비스로 데이터를 푸시하기 시작합니다.

4.1.1 Example of Return-URI Form Posting

아래 예제는 JavaScript를 사용하여 rURI를 추출하고 form.submit ()을 사용하여 양식을 즉시 게시하는 웹 페이지를 사용하여 양식을 return URI (rURI)로 POST하는 방법을 보여줍니다.

```
<script type="text/javascript">
function init () {
    var ruri = "";

    /* Loop through GET parameters sent
    to the control URI (this page). */

    var m = location.search.substr (1).split ("&");
    for (var i = 0; i < m.length; ++i) {
        var nv = m[i].split ("=", 2);
        var name = nv[0];
        var value = nv.length > 1 ? nv[1] : null;

        /* Choose what we should do with
        parameters we acknowledge */
        switch (name) {
           case "ruri": ruri = decodeURIComponent (value); break;
           case "did": devname = decodeURIComponent (value); break;
           case "sn": serialnum = decodeURIComponent (value); break;
           // there's also "rlist", "vlist" and "opts"
        }
    }
    /* We've read the GET parameters, now do something with
    the values. Prepare your database for incoming data from
    the particular device, generate a unique push URI for the
    device, generate credentials, etc. */

    // Generate unique push URI based on device name and
    // set the form value of "uri", which will be the URI the eGauge to
    pushURI = "https://www.example.com/push-data?devname=" + devname;
    document.getElementById("uri").value = pushURI;

    // Finally, submit the form to the return URI
    var form = document.getElementById ("form");
    form.action = ruri;
    form.submit ();
}

window.onload = init;
</script>

<form id="form" method="POST" enctype="multipart/form-data">
  <!-- Note URI is set in JavaScript function -->
  <input type="hidden" name="uri" id="uri"
    value="">
  <!-- You can also set options, user, pw in JavaScript -->
  <input type="hidden" name="options" id="options" value="totals">
  <input type="hidden" name="user" id="user" value="push_username">
  <input type="hidden" name="pw" id="pw" value="push_password">
</form>
```

보다 현실적인 환경에서 위의 웹 페이지는 서버 측 스크립팅 (예 : PHP)에 의해 생성됩니다. 이 경우 반환 URI는 양식에 하드 코딩 될 수 있으며 JavaScript는 양식을 제출하는 데에만 사용해야합니다.

이 예제는 등록 프로세스를 관찰하는 데 사용할 수 있습니다. 텍스트 파일로 저장하고 브라우저에서 열어 적어도 rURI와 이름을 GET 매개 변수로 전달합니다.

예 :http://cURI.html?name=eGauge10010&rURI=http://egauge10010.d.egauge.net/cgi-bin/protected/egauge-cfg?setpush

4.2 Push Service Status Verification

푸시 서비스가 활성화되면 URI 참조에서 상태를 확인할 수 있습니다.

```
/push-status.html
```

이 웹 페이지에는 마지막 푸시 시도의 날짜 및 시간, 시도 중에 수신 된 HTTP 응답 코드, 마지막 성공한 푸시의 날짜 및 시간, 푸시 중에 전달 된 데이터 행의 수와 같은 기본 상태 정보가 표시됩니다.

동일한 정보는 URI 참조시 XML 문서 형식으로도 사용할 수 있습니다.

```
/cgi-bin/get?pushStatus
```

다음 XML 요소는 uploadStatus 요소 내에 반환됩니다.

Element Name:	Type:	Description:
nextAttempt	Integer (U32)	다음에 데이터를 푸시하려고 시도합니다. 이것은 UNIX 시간 소인입니다.
lastAttempt	Integer (U32)	다음에 데이터를 푸시하려고 시도합니다. 이것은 UNIX 시간 소인입니다.
lastSuccess	Integer (U32)	마지막으로 데이터가 성공적으로 푸시되었습니다. 이것은 UNIX 시간 소인입니다
lastUploadCount	Integer (U32)	마지막으로 성공한 푸시 중에 전송 된 데이터 행 수입니다.
lastResponseCode	Integer (U32)	마지막 푸시 시도 중 수신 된 HTTP 응답 코드.
lastTimestamp	Integer (U32)	푸시 된 가장 최근 데이터 행의 타임 스탬프입니다. 유닉스 타임 스탬프입니다

5 Miscellaneous tips and information

5.1 Identifying devices push data

데이터를 푸시하는 장치를 식별하기 위해 푸시 문자열의 끝 부분에 고유 한 토큰이 인코딩 될 수 있습니다. 예를 들어 http://pushdata.example.com/collector/egauge에 데이터 수집기가있는 경우 https://pushdata.example.com/collector/egauge?devid=f6ys3sdrw56754c로 푸시하도록 eGauge를 설정할 수 있습니다. 데이터 수집기는이 GET 매개 변수를 읽고 값에서 푸싱 장치를 결정해야합니다.

참고 : 저장된 데이터 및 순간 데이터의 직렬 속성은 구성이 장치에 변경 될 때 변경됩니다. 따라서 장치의 고유 한 표시기로 사용할 수 없습니다.