Luminea-Protokoll

Verwendet wird hier das TUYA-Protokoll, der nicht sehr gut dokumentiert ist. Kommuniziert wird entweder über den zentralen Server oder direkt, dann über Port 6668 TCP. Von jeder Komponente müssen folgende 3 Komponenten bekannt sein:

- IP (kann über die App abgefragt werden)
- ID (kann über die App abgefragt werden) die '01200???' dann 12 Zeichen MAC ohne Trennzeichen
- localKey der Verschlüsselungskey
  (Bestimmung des Keys siehe hier:LumineaLokalKeyBestimmen)
  Ohne den Key kann der Status gepollt werden, aber es kann nicht gesteuert werden und man kann auch keine autom. Nachrichten bei Änderung empfangen

Protokollbeschreibung

Jedes Datenpaket ist folgendermaßen aufgebaut:

von	bis	bytes	Name	Beschreibung	Beispiel
0	10	11	<prefix>	Telegrammstart (fix)	00 00 55 AA 00 00 00 00 00 00 00
11	11	1	<cmd>	command	08
12	15	4	<len>	Länge bis zum Telegrammende ab byte 16 incl suffix	00 00 00 8b = 139 bytes 16..155
16	19	4	<?>	immer 0	00 00 00 00
20	<len>+7	11	<rawdata>	Rohdaten	
<len>+8	<len>+11	4	<magic>	keine Ahnung was das bedeutet, beim senden immer 0 setzen	02 4a 3e 07
<len>+12	<len>+15	4	<suffix>	Telegrammende (fix)	00 00 aa 55

Die Rohdaten können verschlüsselt sein oder nicht. Die mit STATUS unverschlüsselt angefragten Werte werden auch unverschlüsselt beantwortet, die automatischen Änderungsmeldungen RESPONSE waren bisher verschlüsselt. Wenn die Rohdaten mit der Versionsnumer beginnen (momentan 3.1, d.h. die Bytes 33 2e 31), dann ist der Rest verschlüsselt und mus erst entschlüsselt werden. Ansonsten wird dieser Schritt übersprungen

Entschlüsselung

Wenn die Rohdaten verschlüsselt sind, dann sind sie folgendermaßen aufgebaut:

von	bis	bytes	Name	Beschreibung	Beispiel	
0	2	3	<Version>	Versionsnummer (fix)	33 2e 31	
3	18	16	<md5>	Hashwert	Dies sind die Bytes 8 bis 16 des md5 Hash über folgenden String: "data={dataEncB64}||lpv={Version}||{localKey}"	
19			<dataEncB64>	verschlüsselte Daten Base 64codiert		

Als nächstes müssen die Daten aus der Base64 Dastellung in einen String zurückkonvertiert werden:
dataEnc = Convert.FromBase64String(dataEncB64)

Anschließend können die Daten mit dem localKey entschlüsselt werden (AES, CipherMode = ECB) (Bestimmung des Keys siehe hier:LumineaLokalKeyBestimmen)

Dann erhält man einen json String in folgendem Format (Beispiel ZX2820):

```
{"devId":"01200xxxxxxxxxxxxxx","dps":{"1":true,"2":0,"4":47,"5":56,"6":2328}}
{"devId":"01200xxxxxxxxxxxxxx","dps":{"4":48,"5":59},"t":1516641459,"s":21770}
```

Die erste Zeile ist die Antwort auf einen Request, diese wir unverschlüsselt gesendet.
Die zweite wird automatisch gesendet sobald sich Daten ändern.

Key	Bedeutung	Beschreibung
devId	Device Id	Eindeutige Kennnummer des Gerätes
dps	Datenpunkt	dieser ist selbst ein json String und enthält die eigentlichen Werte diese sind abhängig vom Gerät, siehe unten
t	Zeit	Unix Zeit Nur bei automatisch gesendeten Werten
s	Sequenznummer	fortlaufende Nummer der automatisch gesendeten Telegramme Nur bei automatisch gesendeten Werten

Datenwerte

ZX2820

Nummer	ZX2832	ZX2831	ZX2820	
1	powered	powered	powered	
2	mode	brightnes	delay	
3	brightnes	colTemp	temp	
4	colTemp		urrent	
5	colValue		power	
6	scenePara		voltage	
7	scene_1Para			
8	scene_2Para			
9	scene_3Para			
10	scene_4Para			