

# HTTP Uygulama Notu

Web sitemize bağlanın ve herhangi bir yardım için teknik destek ekibimizle iletişime geçmekten çekinmeyin.

**Cavli Inc.,**

99 South Almaden Blvd., Suite 600, San Jose, California, 95113

Tel: 1-650-535-1150

Web: [www.cavliwireless.com](http://www.cavliwireless.com)

IoT Connectivity Platform: [www.cavlihubble.io](http://www.cavlihubble.io)

**Destek Merkezi**

<https://www.cavliwireless.com/support-center.html>

e-Mail: [support@cavliwireless.com](mailto:support@cavliwireless.com)

**Satış sorularınız için**

<https://www.cavliwireless.com/contact-us.html>

e-Mail: [sales@cavliwireless.com](mailto:sales@cavliwireless.com)

**Daha Fazla IoT Modülü**

<https://www.cavliwireless.com/iot-modules/cellular-modules.html>

**COPYRIGHT**

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF CAVLI WIRELESS INC.

TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF

DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL

OR DESIGN. EVERY EFFORT HAS BEEN MADE IN PREPARATION OF THIS DOCUMENT TO ENSURE ACCURACY OF THE

CONTENTS. BUT ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT DO NOT

CONSTITUTE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. THE INFORMATION IN THIS DOCUMENT IS SUBJECT

TO CHANGE WITHOUT NOTICE DUE TO PRODUCT VERSION UPDATE OR OTHER REASONS. FOR MOST RECENT

DOCUMENTS, ALWAYS REFER THE PRODUCT PORTFOLIO SECTION AT

[www.cavliwireless.com](http://www.cavliwireless.com)

**Copyright © Cavli Inc. All rights reserved**

**Sürüm Tarihçesi**

<b>Sürüm</b>	<b>Düzenleme</b>	<b>Tarih</b>
<b>1.0</b>	İlk Sürüm	31/01/2024

## İçindekiler

1. Giriş.....	6
2. Ön Hazırlık.....	7
2.1. EVK'nın PC'ye bağlanması.....	7
3. HTTP POST/PUT Uygulaması.....	9
3.1. Geçerli HTTP oturumunun URL'sini ayarlamak.....	9
3.2. Header alanının HTTP istek alanına eklenmesi.....	9
3.3. İçeriğin güncellenmesi.....	9
3.4. POST isteğinin gerçekleştirilmesi.....	10
4. HTTP GET Uygulaması.....	10
4.1. Geçerli HTTP oturumunun URL'sini ayarlamak.....	10
4.2. Header alanının HTTP istek alanına eklenmesi.....	11
4.3. GET isteğinin gerçekleştirilmesi.....	11
4.4. HTTP isteğinin header yanıtının alınması.....	11
4.5. HTTP isteğinin yanıt durumunun alınması.....	12
4.6. İçerik uzunluğunun alınması.....	13
4.7. İçeriğin alınması.....	13

## Resimler

Resim 1: EVK AT USB Port.....	7
Resim 2: EVK PowerKey ve STS Led'i.....	8
Resim 3: POST/PUT uygulaması için HTTP içeriği.....	9
Resim 4: HTTP İçeriği.....	10

# 1. Giriş

Bu belge, Cavli C16QS Geliştirme Kiti (EVK) kullanılarak HTTP uygulamasını anlatmaktadır.

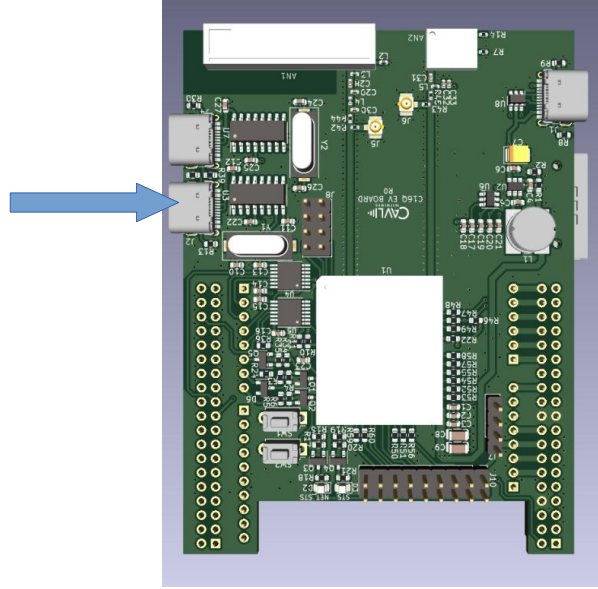
**Not:**

- *Uygulama AT UART portu olan UART 1 üzerinden yapılmıştır.*

## 2. Ön Hazırlık

### 2.1. EVK'nın PC'ye bağlanması

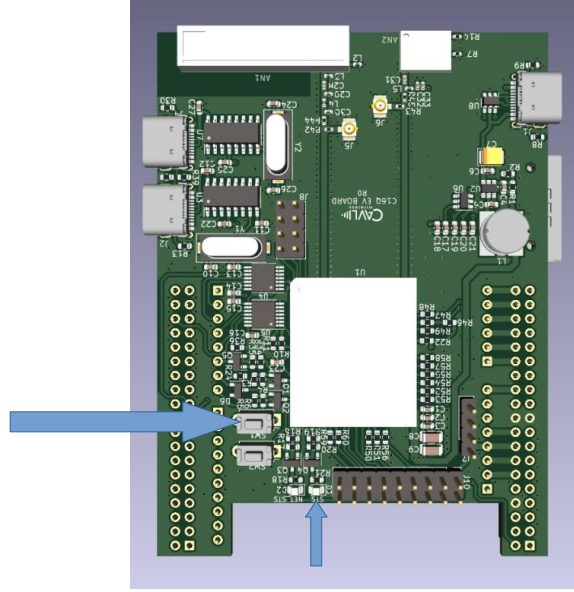
- USB Type C data kablosunu bilgisayarınızın USB portlarından birine bağlayın.
- USB Type C data kablosunun diğer ucunu EVK kartınızın AT komutu için kullanılacak olan USB konnektörüne takın.



Resim 1: EVK AT USB Port

- EVK kartı enerjisini USB üzerinden almaktadır.
- Windows işletim sistemi COM port olarak, Linux işletim sistemi ise ttyUSB olarak EVK kartınızı otomatik tanıyacaktır.
- İşletim sisteminizde herhangi bir terminal programını açın ve ayarlarınızı aşağıdaki gibi yapılandırın:
  - Baud Rate : 115200
  - Data Bits : 8
  - Stop Bits : 1
  - Parity : None

- EVK kartı üzerinde yer alan **SW1(POWERKEY)** butonuna kısa süreli basın ve bırakın. Kart üzerinde bulunan Yeşil renk **STS** ledi on durumuna gelecektir.



Resim 2: EVK PowerKey ve STS Led'i

- Terminal programınızda aşağıdaki mesaj gelecektir:

**^boot.rom'v'!\n**

**RDY**

**+ATREADY**

**+CAVEUICCSUPPORT:0**

Onaltılık gösterim;

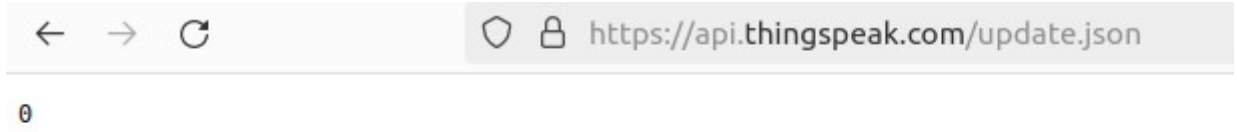
```
00000000 5E 62 6F 6F 74 2E 72 6F  6D 01 27 76 00 00 00 02
00000010 27 21 5C 6E 0D 0A 52 44  59 0D 0A 0D 0A 2B 41 54
00000020 52 45 41 44 59 0D 0A 0D  0A 2B 43 41 56 45 55 49
00000030 43 43 53 55 50 50 4F 52  54 3A 30 20 0D 0A
```

```
^boot.rom.'v....
'!\n..RDY....+AT
READY....+CAVEUI
CCSUPPORT:0 ..
```



### 3. HTTP POST/PUT Uygulaması

Uygulama için örneğe verilen HTTP adresinde bulunan veriler aşağıdaki gibidir.



Resim 3: POST/PUT uygulaması için HTTP içeriği

#### 3.1. Geçerli HTTP oturumunun URL'sini ayarlamak

Geçerli HTTP oturumunun URL adresini ayarlamak için AT komut portu üzerinden aşağıdaki komut gönderilir.

**AT+HTTPURL="https://api.thingspeak.com/update.json"<CR><LF>**

ASCII formatta gelen cevap:

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 4F 4B 0D 0A ....OK..

#### 3.2. Header alanının HTTP istek alanına eklenmesi

Header alanının HTTP istek alanına eklenmesi için AT komut portu üzerinden aşağıdaki komut gönderilir.

**AT+HTTPADDHEAD="Content-Type: application/json"<CR><LF>**

ASCII formatta gelen cevap:

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 4F 4B 0D 0A ....OK..

#### 3.3. İçeriğin güncellenmesi

HTTP içeriğinin güncellenmesi için AT komut portu üzerinden aşağıdaki komut gönderilir.

**AT+HTTPCONTENT=2<CR><LF>**

Daha sonra veri girişi için ">" karakteri cevap olarak döner.

ASCII formatta gelen cevap:

**>**

**12 sayısını girelim.**

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 3E 20 0D 0A 4F 4B 0D 0A ....>..OK..

### 3.4. POST isteğinin gerçekleştirilmesi

Güncelenen içeriğin POST fonksiyonu ile gönderilmesi için AT komut portu üzerinden aşağıdaki komut gönderilir.

**AT+HTTPREQUEST=POST/PUT<CR><LF>**

ASCII formatta gelen cevap:

**HTTPSEND:SUCCESS**

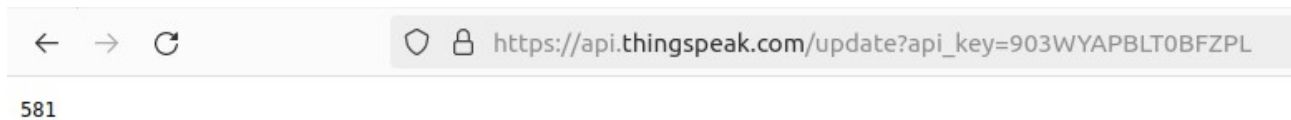
**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 0D 0A 48 54 54 50 53 45 4E 44 3A 53 .....HTTPSEND:S  
00000010 55 43 43 45 53 53 0D 0A 0D 0A 4F 4B 0D 0A UCCCESS....OK..

## 4. HTTP GET Uygulaması

Uygulama için örnekte verilen HTTP adresinde bulunan veriler aşağıdaki gibidir. Bu web sayfasında bulunan içerik 581 verisidir. GET uygulaması ile bu içeriği Cavli C16QS üzerinden alan uygulamayı anlatacağız.



Resim 4: HTTP İçeriği

### 4.1. Geçerli HTTP oturumunun URL'sini ayarlamak

Geçerli HTTP oturumunun URL adresini ayarlamak için AT komut portu üzerinden aşağıdaki komut gönderilir.

**AT+HTTPURL="https://api.thingspeak.com/update?api\_key=903WYAPBLT0BFZPL"<CR><LF>**

ASCII formatta gelen cevap:

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 4F 4B 0D 0A ....OK..

## 4.2. Header alanının HTTP istek alanına eklenmesi

Header alanının HTTP istek alanına eklenmesi için AT komut portu üzerinden aşağıdaki komut gönderilir.

**AT+HTTPADDHEAD="Content-Type:application/x-www-form-urlencoded"<CR><LF>**

ASCII formatta gelen cevap:

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 0D 0A 4F 4B 0D 0A ....OK..

Örnek uygulama için ikinci alanı da gönderiyoruz.

**AT+HTTPADDHEAD="Accept: \*/\*"<CR><LF>**

ASCII formatta gelen cevap:

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 0D 0A 4F 4B 0D 0A ....OK..

## 4.3. GET isteğinin gerçekleştirilmesi

HTTP sunucusuna GET isteği AT komut portu üzerinden aşağıdaki komut gönderilir. Kısa bir beklemenin ardından istek olumlu karşılanmış ise **SUCCESS** olarak geri dönüş sağlanır.

**AT+HTTPREQUEST=GET<CR><LF>**

ASCII formatta gelen cevap:

**HTTPSEND:SUCCESS**

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 0D 0A 48 54 54 50 53 45 4E 44 3A 53 .....HTTPSEND:S  
00000010 55 43 43 45 53 53 0D 0A 0D 0A 4F 4B 0D 0A UCCCESS....OK..

## 4.4. HTTP isteğinin header yanıtının alınması

HTTP isteğinin header yanıtının alınması için AT komut portu üzerinden aşağıdaki komut gönderilir. Aşağıdaki gibi bir yanıt alınır.

**AT+HTTPGETHEAD?<CR><LF>**

ASCII formatta gelen cevap:

**X-Frame-Options: SAMEORIGIN**

**ETag: W/"de0023e398111d43424845aaeee2e119"**

**Access-Control-Allow-Methods: GET, POST, PUT, OPTIONS, DELETE, PATCH**

**Access-Control-Allow-Headers: origin, content-type, X-Requested-With**

**X-Request-Id: 58566995-28ec-4549-822a-9d75e31b51aa**

**Access-Control-Max-Age: 1800**

**Access-Control-Allow-Origin: \***

**Cache-Control: max-age=0, private, must-revalidate**

Status: 200 OK  
Connection: keep-alive  
Content-Length: 3  
Content-Type: text/plain; charset=utf-8  
Date: Tue, 30 Jan 2024 08:55:09 GMT  
HTTP/1.1 200 OK

OK

Onaltılık formatta gelen cevap:

```
00000000 0D 0A 0D 0A 58 2D 46 72 61 6D 65 2D 4F 70 74 69 ....X-Frame-Opti
00000010 6F 6E 73 3A 20 53 41 4D 45 4F 52 49 47 49 4E 0D ons: SAMEORIGIN.
00000020 0A 45 54 61 67 3A 20 57 2F 22 64 65 30 30 32 33 .ETag: W/"de0023
00000030 65 33 39 38 31 31 31 64 34 33 34 32 34 38 34 35 e398111d43424845
00000040 61 61 65 65 65 32 65 31 31 39 22 0D 0A 41 63 63 aae2e119"...Acc
00000050 65 73 73 2D 43 6F 6E 74 72 6F 6C 2D 41 6C 6C 6F ess-Control-Allo
00000060 77 2D 4D 65 74 68 6F 64 73 3A 20 47 45 54 2C 20 w-Methods: GET,
00000070 50 4F 53 54 2C 20 50 55 54 2C 20 4F 50 54 49 4F POST, PUT, OPTIO
00000080 4E 53 2C 20 44 45 4C 45 54 45 2C 20 50 41 54 43 NS, DELETE, PATC
00000090 48 0D 0A 41 63 63 65 73 73 2D 43 6F 6E 74 72 6F H..Access-Contro
000000a0 6C 2D 41 6C 6C 6F 77 2D 48 65 61 64 65 72 73 3A l-Allow-Headers:
000000b0 20 6F 72 69 67 69 6E 2C 20 63 6F 6E 74 65 6E 74 origin, content
000000c0 2D 74 79 70 65 2C 20 58 2D 52 65 71 75 65 73 74 -type, X-Request
000000d0 65 64 2D 57 69 74 68 0D 0A 58 2D 52 65 71 75 65 ed-With..X-Reque
000000e0 73 74 2D 49 64 3A 20 35 38 35 36 36 39 39 35 2D st-Id: 58566995-
000000f0 32 38 65 63 2D 34 35 34 39 2D 38 32 32 61 2D 39 28ec-4549-822a-9
0000100 64 37 35 65 33 31 62 35 31 61 61 0D 0A 41 63 63 d75e31b51aa..Acc
0000110 65 73 73 2D 43 6F 6E 74 72 6F 6C 2D 4D 61 78 2D ess-Control-Max-
0000120 41 67 65 3A 20 31 38 30 30 0D 0A 41 63 63 65 73 Age: 1800..Acces
0000130 73 2D 43 6F 6E 74 72 6F 6C 2D 41 6C 6C 6F 77 2D s-Control-Allow-
0000140 4F 72 69 67 69 6E 3A 20 2A 0D 0A 43 61 63 68 65 Origin: *.Cache
0000150 2D 43 6F 6E 74 72 6F 6C 3A 20 6D 61 78 2D 61 67 -Control: max-ag
0000160 65 3D 30 2C 20 70 72 69 76 61 74 65 2C 20 6D 75 e=0, private, mu
0000170 73 74 2D 72 65 76 61 6C 69 64 61 74 65 0D 0A 53 st-revalidate..S
0000180 74 61 74 75 73 3A 20 32 30 30 20 4F 4B 0D 0A 43 tatus: 200 OK..C
0000190 6F 6E 6E 65 63 74 69 6F 6E 3A 20 6B 65 65 70 2D onnection: keep-
00001a0 61 6C 69 76 65 0D 0A 43 6F 6E 74 65 6E 74 2D 4C alive..Content-L
00001b0 65 6E 67 74 68 3A 20 33 0D 0A 43 6F 6E 74 65 6E ength: 3..Conten
00001c0 74 2D 54 79 70 65 3A 20 74 65 78 74 2F 70 6C 61 t-Type: text/pla
00001d0 69 6E 3B 20 63 68 61 72 73 65 74 3D 75 74 66 2D in; charset=utf-
00001e0 38 0D 0A 44 61 74 65 3A 20 54 75 65 2C 20 33 30 8..Date: Tue, 30
00001f0 20 4A 61 6E 20 32 30 32 34 20 30 38 3A 35 35 3A Jan 2024 08:55:
0000200 30 39 20 47 4D 54 0D 0A 48 54 54 50 2F 31 2E 31 09 GMT..HTTP/1.1
0000210 20 32 30 30 20 4F 4B 0D 0A 0D 0A 4F 4B 0D 0A 200 OK....OK..
```

## 4.5. HTTP isteğinin yanıt durumunun alınması

HTTP isteğinin yanıt durumunun alınması için AT komut portu üzerinden aşağıdaki komut gönderilir. Aşağıdaki gibi bir yanıt alınır.

AT+HTTPGETSTAT?<CR><LF>

ASCII formatta gelen cevap:

STATUS\_RESPONSE:200

OK

Onaltılık formatta gelen cevap:

```
00000000 0D 0A 0D 0A 53 54 41 54 55 53 5F 52 45 53 50 4F ....STATUS_RESPO
00000010 4E 53 45 3A 32 30 30 0D 0A 0D 0A 4F 4B 0D 0A NSE:200....OK..
```

## 4.6. İçerik uzunluğunun alınması

İçerik uzunluğunun alınması için AT komut portu üzerinden aşağıdaki komut gönderilir. Aşağıdaki gibi bir yanıt alınır.

**AT+HTTPGETCLEN?<CR><LF>**

ASCII formatta gelen cevap:

**CONTENT\_LENGTH:3**

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 0D 0A 43 4F 4E 54 45 4E 54 5F 4C 45 4E 47 ....CONTENT LENG  
00000010 54 48 3A 33 0D 0A 0D 0A 4F 4B 0D 0A TH:3....OK..

## 4.7. İçeriğin alınması

İçeriğin alınması için AT komut portu üzerinden aşağıdaki komut gönderilir. Aşağıdaki gibi bir yanıt alınır.

**AT+HTTPGETCONT?<CR><LF>**

ASCII formatta gelen cevap:

**581**

**OK**

Onaltılık formatta gelen cevap:

00000000 0D 0A 35 38 31 0D 0A 4F 4B 0D 0A ..581..OK..