

USB 2..24 KANAL RÖLE KARTI

Kullanım Kılavuzu

Model: USB_x_Out

Firmware Sürümü: FW v1.0
Yazılım Sürümü: USB_x_Out v1.0
Belge Sürümü: 1.1

dELAb - Deniz Elektronik Laboratuvarı

1. Genel Tanıtım

USB_x_Out röle kartı, bir bilgisayar veya gömülü sistem üzerinden 24 adede kadar bağımsız röleyi kontrol etmenizi sağlayan USB tabanlı bir yardımcı modüldür. Kart, USB CDC sınıfında çalıştığı için işletim sistemine sanal bir COM portu olarak görünür ve ek sürücü gerektirmez (Linux ve macOS'ta tak-çalıştır; Windows 10/11'de otomatik tanınır).

1.1 Tipik Kullanım Alanları

- Aydınlatma kontrol panelleri
- Test ve ölçü otomasyonları
- Eğitim laboratuvarı deneyleri
- Endüstriyel proses kontrol arayüzleri
- Akıllı bina otomasyon arayüzleri
- DIY ev otomasyonu

NOT

USB Röle kontrol kartının kullanım testi ve kendi projelerinize kodlama örneği olarak kullanmak üzere hazırladığımız USB_x_Out yazılımını kaynak kodlarıyla web sitemizden indirebilirsiniz.

2. Güvenlik Uyarıları

⚠ UYARI

Şebeke gerilimiyle çalışırken röle bankası ile USB kontrol kartı arasında galvanik izolasyon mutlaka sağlanmalıdır. Standart opto-izoleli röle modülleri bunu sağlar; ancak izolasyonu kontrol etmek son kullanıcının sorumluluğundadır.

⚠ UYARI

2 ve 4 kanallı kontrol kartlarında harici besleme gerekmez, besleme USB üzerinden alınır. Daha yüksek sayıda çıkış olan modüllerde karta verilen 5 V güç beslemesi regüleli ve polarite koruma altında olmalıdır. Yanlış polarite veya 6 V üstü gerilimler kalıcı hasara yol açabilir.

⚠ UYARI

2 ve 4 kanallı kontrol kartlarında harici besleme gerekmez, besleme USB üzerinden alınır. Daha yüksek sayıda çıkışa sahip kartlarda röle bobinlerini USB hattından beslemeyin. $24 \text{ bobin} \times \sim 70 \text{ mA} = 1.7 \text{ A}$ olabilir; USB 2.0 portu bunu sağlamaz.

⚠ UYARI

Statik elektrik hassasiyeti: Karta dokunmadan önce topraklı bir yüzeye temas edin. PCB üzerindeki açık komponentler ESD'ye duyarlıdır.

3. İlk Kurulum ve Çalıştırma

3.1 Donanım Kurulumu

1. Karta 5 V güç adaptörünü bağlayın (2 ve 4 çıkışlı kartlarda gerekmez) (J1).
2. Röle bankasını J4'e bağlayın.
3. USB kablosuyla kartı bilgisayara bağlayın (J2).

3.2 Açılış Davranışı

- Kart fişe takılır takılmaz açılır.
- Tüm röleler **kapalı** durumda başlar.
- USB bilgisayara takıldığında işletim sistemi yeni bir COM portu (örn. COM4) tanır.

3.3 İlk Test (USB_x_Out yazılımı)

4. USB24RelayControl.exe programını çalıştırın.
5. Cihazın bağlı olduğu COM portu (★) ile işaretlenmiştir, bu portu seçin.
6. **Bağlan** butonuna tıklayın. Durum LED'i yeşile döner.
7. **TUM01** butonuna basın → tüm röleler kapalıdan açığa geçmelidir.
8. **TUM00** butonuna basın → tüm röleler kapanmalıdır.
9. Tek tek kutucuklara tıklayarak bağımsız kontrolü test edin.

-  **AÇ/KAPA (Otomatik Döngü)** → her saniyede bir TUM01/TUM00 komutu gönderir; voltaj/akım testi için. Tekrar tıklayınca durur.

****Alt sıra:****

- **TUM01** → 24 röleyi aynı anda aç
- **TUM00** → 24 röleyi aynı anda kapat
- **DRM01** → 1. röleyi sorgula (örnek)
- **TGL01** → 1. röleyi tersle (örnek)
- **HEX** → 24-bit hex maskeyle toplu ayar (kullanıcı pencerede maskeyi girer)
- **BIN** → 24-bit binary maskeyle toplu ayar

4.3 Önerilen Çalışma Akışı

10. Karta güç verin, USB kablosunu takın.
11. USB_x_Out'u açın, yıldızlı portu seçip bağlanın.
12. **STA** butonuyla durumu kontrol edin.
13. Senaryoyu uygulayın (tekil tıklama veya HEX/BIN maske).
14. Çalışmanız bittiğinde **TUM00** ile temizleyin, sonra **Bağlantıyı Kes**.

5. Komut Seti — Tam Referans

Bu bölüm, kendi yazılımını yazmak isteyen geliştiriciler ve doğrudan terminal üzerinden çalışan kullanıcılar içindir.

5.1 Genel Kurallar

- Tüm komutlar ASCII'dir.
- Komut sonuna \n veya \r\n ekleyin (Enter).
- Komutlar büyük/küçük harf duyarlıdır; cihaz büyük harfe çevirir.
- Maksimum komut uzunluğu **63 karakter**; aşılsa HTA: TASMA döner.
- Cihaz her komuta bir cevap satırı gönderir (yardım komutu hariç — çok satır).
- Cevaplar \r\n ile biter.

5.2 Komut Tablosu

Komut	Açıklama	Örnek	Cevap
ACCnn	n. röleyi aç (nn = 01..24)	ACC07	TMM
KPAnn	n. röleyi kapat	KPA07	TMM
TGLnn	n. röleyi tersle	TGL07	TMM
DRMnn	n. röle durumunu sorgula	DRM07	DRM:07:1
TUM00	tüm röleleri kapat	TUM00	TMM
TUM01	tüm röleleri aç	TUM01	TMM
STAxx	tam durum sorgula (xx göz ardı)	STA00	STA:00000000000000000000100
STA	aynı, kısa biçim	STA	STA:00000000000000000000100
HEXh...	hex maske ile toplu ayar (24 bit)	HEX800001	TMM
BINb...	binary maske ile toplu ayar (24 bit)	BIN1000...0001	TMM
VER	firmware sürümü	VER	VER:24R-1.0
YRD	yardım metni	YRD	(çok satır)
RST	tüm röleleri kapat + reset	RST	TMM (sonra USB kopar)

5.3 Cevap Formatları

Cevap	Anlam
TMM	Komut başarılı (Tamam)
HTA:SOZDIZIM	Sözdizim hatası
HTA:NUMARA	Röle numarası 1..24 dışında
HTA:BILINMEYEN	Tanınmayan komut
HTA:TASMA	Komut tamponu taşması

Cevap	Anlam
STA:bbbbbb...bbbb	24 karakter binary; MSB = RLY24, LSB = RLY1
HEX:hhhhhh	6 karakter büyük harf hex
DRM:nn:s	nn = röle no, s = 0 (kapalı) veya 1 (açık)
FW v1.0	Sürüm bilgisi

5.4 Bit Sıralaması

Bu bölüm kritiktir. HEX, BIN ve STA cevabında bit numaralandırması aynıdır:

bit 23 bit 22 bit 21 ... bit 2 bit 1 bit 0
RLY24 RLY23 RLY22 ... RLY3 RLY2 RLY1

↑
en SOLDAKİ karakter

↑
en SAĞDAKİ karakter

****Örnekler:****

Komut	Sonuç
HEX000001	sadece RLY1 açık (00000000000000000000 1)
HEX000002	sadece RLY2 açık (00000000000000000000 10)
HEX800000	sadece RLY24 açık (1 0000000000000000000000)
HEX800001	RLY1 ve RLY24 açık (1 000000000000000000000 1)
HEXFFFFFF	hepsi açık (TUM01 ile aynı)
HEX000000	hepsi kapalı (TUM00 ile aynı)
HEX00FF00	RLY9..RLY16 açık
BIN11111111111111111111111111111111	hepsi açık
BIN10000000000000000000000001	RLY24 ve RLY1 açık (1 0000000000000000000000 1)

`STA:` cevabı da aynı sırada gelir:

STA:10000000000000000000000001
 ↑ ↑
 RLY24 açık RLY1 açık

5.5 Açılış Mesajları

Cihaz USB takıldığında veya reset sonrası şu satırları yollar (ihtiyari, kullanıcı yazılımı bunları parse etmek zorunda değil):

```
dELab USB_X_Out FW 1.0
YRD ile yardim
STA:00000000000000000000000000000000
```

6. Terminal Üzerinden Kullanım

Kendi programınızı yazmıyor ama terminal/serial monitor ile manuel kontrol etmek istiyorsanız:

6.1 Önerilen Programlar

- **PuTTY** (Windows) — Serial, COM portu, 115200 baud
- **Tera Term** (Windows) — daha gelişmiş seri monitor
- **screen** (Linux/macOS) — `screen /dev/ttyACM0 115200`
- **minicom** (Linux/macOS) — `minicom -D /dev/ttyACM0 -b 115200`

6.2 Seri Port Ayarları

Parametre	Değer
Baud rate	115200
Data bits	8
Parity	None
Stop bits	1
Flow control	None
Encoding	ASCII
Newline	CR+LF (\r\n) — gönderme için LF de yeterli

6.3 Örnek Tur

Bağlandıktan sonra şu komutları sırayla yazın (her komut sonunda Enter):

```

VER
→ VER:FW v1.0

TUM00
→ TMM

ACC01
→ TMM

DRM01
→ DRM:01:1

ACC24
→ TMM

STA
→ STA:10000000000000000000000000000001

HEX800001
→ TMM

TUM00
→ TMM

```

7. Sorun Giderme

Belirti	Olası Neden	Çözüm
Listede COM port görünmüyor	USB kablosu sadece şarj kablosu	Veri destekli USB kablosu kullan
	Karta güç gelmiyor	5 V adaptörü kontrol et
	USB sürücü problemi (Windows)	Cihaz Yöneticisi'nden manuel olarak VCP sürücüsünü güncelle
COM portu görünüyor ama bağlantı kurulmuyor	Port başka programda açık	Serial Monitor, PuTTY, vb. terminali kapat
Komutu gönderdim ama cevap gelmiyor	Enter (CR/LF) atlanmış	Terminal ayarlarında "Send CR+LF on Enter" aktif olduğunu kontrol et
Tüm röleler birlikte çekilince güç düşüyor	Adaptör yetersiz	5 V / 2 A veya üstü adaptör kullan
Bazı röleler tetiklenmiyor	J4 kablo bağlantısı	Konnektör pinlerini gözden geçir, oksidasyon olmadığından emin ol
HTA:NUMARA cevabı	nn değeri 01..24 dışında	Sıfır dolgulu iki haneli yaz: ACC07
HTA:SOZDIZIM cevabı	HEX maske 6 haneden fazla	En fazla FFFFFFFF (= 24 bit) gönderebilirsin
HTA:BILINMEYEN cevabı	Komut yanlış yazılmış	Büyük/küçük harf önemsiz, boşluk olmamalı
Karta dokununca ölçüm değişiyor	Statik elektrik	ESD önlemleri al, kart kasa içine yerleştirilmeli

8. Teknik Ekler

8.1 Komut/Cevap Atomikliği

- Tek komutlar atomic değildir — ACC01 çağrısı yalnız RLY1'i etkiler.
- TUM01, TUM00, HEX. . . , BIN. . . komutları **24 röleyi aynı anda** günceller; arada başka bir durum yoktur.
- Bu nedenle "16 röleyi sırayla aç" senaryosu yerine HEX00FFFF tek komutu daha hızlı ve atomic olarak çalışır.

8.2 USB Re-Enumerate Davranışı

`RST` komutu cihazı resetler ve dolayısıyla USB hattını koparır ve yeniden bağlanır:

- Önce TMM cevabı gelir.
- ~50 ms sonra USB hattı kapanır (Windows'ta cihaz "kaldırıldı" sesi gelebilir).
- ~1 saniye sonra cihaz yeniden enumerate olur ve aynı COM port adıyla geri gelir.

USB_x_Out yazılımı bu durumu otomatik algılar; manuel yeniden bağlanma gerekmez.

8.3 Maksimum İletişim Hızı

- Komut tamponu 63 karakter
- Tipik komut süresi: ~5 ms (komut alma + işleme + cevap gönderme)
- Saniyede ~200 komut güvenle gönderilebilir; daha hızlı tempo HTA:TASMA riski getirir.

9. Farklı Diller İçin Örnek Kodlar

9.1 Python Entegrasyonu (pyserial kullanarak)

```
```python
import serial
import time

class RelayController:
 def __init__(self, port_name, baudrate=115200):
 # DTR/RTS'yi kapalı tutmak önemlidir
 self.ser = serial.Serial(port=port_name, baudrate=baudrate, timeout=1.0)
 self.ser.dtr = False
 self.ser.rts = False

 # Açılış banner'ını temizle
 time.sleep(0.2)
 self.ser.reset_input_buffer()

 def send_cmd(self, cmd):
 self.ser.reset_input_buffer()
 self.ser.write(f'{cmd}\n'.encode('ascii'))
 response = self.ser.readline().decode('ascii').strip()
 return response

 def turn_on(self, relay_no):
 # Örnek: relay_no = 7 -> ACC07
 resp = self.send_cmd(f'ACC{relay_no:02d}')
 if resp != "TMM":
 raise Exception(f'Röle açılmadı: {resp}')

 def get_all_states(self):
 resp = self.send_cmd("STA")
 if resp.startswith("STA:"):
 state_str = resp[4:] # 24 haneli 01 string'i
 # bit 0 (en sağda) = Röle 1 durumudur. Bu yüzden listeyi ters çevirebiliriz
 states = [char == '1' for char in reversed(state_str)]
 return states # states[0] = Röle 1 durumu
 raise Exception("Durum okunamadı")

 def close(self):
 self.ser.close()
```

```
Kullanım Örneği:
controller = RelayController("COM4")
controller.turn_on(1)
print(controller.get_all_states())
controller.close()
'''
```

## 9.2 C# (.NET) Entegrasyonu

```
```csharp
using System;
using System.IO.Ports;
using System.Threading;

public class RelayController : IDisposable
{
    private readonly SerialPort _port;
    private readonly object _lock = new object();

    public RelayController(string portName)
    {
        _port = new SerialPort(portName, 115200, Parity.None, 8, StopBits.One)
        {
            Encoding = System.Text.Encoding.ASCII,
            NewLine = "\r\n",
            ReadTimeout = 1000,
            WriteTimeout = 1000,
            DtrEnable = false,
            RtsEnable = false
        };
    }

    public void Open()
    {
        _port.Open();
        _port.DiscardInBuffer();
        Thread.Sleep(200); // Banner'ın tamamlanması için bekleme
        _port.DiscardInBuffer(); // Gelen banner verisini çöpe at
    }

    public string SendCommand(string command)
    {
    }
```

```

    lock (_lock)
    {
        if (!_port.IsOpen) throw new InvalidOperationException("Port kapalı!");
        _port.DiscardInBuffer();
        _port.WriteLine(command);

        // Yanıt satırını oku
        return _port.ReadLine().Trim();
    }
}

```

```

public void SetRelay(int relayNo, bool state)
{
    if (relayNo < 1 || relayNo > 24) throw new ArgumentOutOfRangeException();
    string cmd = (state ? "ACC" : "KPA") + relayNo.ToString("D2");
    string resp = SendCommand(cmd);
    if (resp != "TMM") throw new Exception($"Komut başarısız oldu: {resp}");
}

```

```

public void Dispose()
{
    _port?.Dispose();
}

}

...

```

C. Node.js (serialport kullanarak)

```

```javascript
const { SerialPort } = require('serialport');
const { ReadlineParser } = require('@serialport/parser-readline');

const port = new SerialPort({
 path: 'COM4',
 baudRate: 115200,
 rtscts: false,
 dsrdtr: false // DTR/RTS pasif
});

```

```

const parser = port.pipe(new ReadlineParser({ delimiter: '\r\n' }));

```

```

// İlk bağlantıda gelen banner'ı yoksaymak için bayrak

```

```

let isReady = false;

```

```
setTimeout(() => {
 isReady = true;
 console.log("Cihaz iletişimine hazır.");
}, 300);

parser.on('data', (data) => {
 if (isReady) {
 console.log(`Cihazdan Gelen: ${data.trim()}`);
 } else {
 console.log(`Açılış Logu: ${data.trim()}`);
 }
});

// Röle 5'i açma komutu gönder
function turnOnRelay5() {
 port.write("ACC05\n", 'ascii', (err) => {
 if (err) return console.log('Hata: ', err.message);
 console.log('Komut gönderildi: ACC05');
 });
}
...

```

## 10. Yazılım Geliştiriciler İçin Hata Giderme ve İpuçları

1. **\*\*İplik Güvenliği (Thread-Safety):\*\*** `SerialPort` nesneleri thread-safe (iplik güvenli) değildir. UI iş parçacığı ile polling (durum sorgulama) iş parçacığının çakışmaması için gönderim fonksiyonlarınızı bir kilit (`lock` veya semafor) mekanizması ile koruyun.
2. **\*\*Kopma ve Yeniden Bağlanma (Reconnection):\*\*** Kartın gücü kesilirse veya USB kablosu çekilirse seri port nesnesi `IOException` fırlatır. Bu durumda mevcut port nesnesini `Dispose` edin, 1 saniye bekleyip portları tekrar tarayarak otomatik bağlanmayı deneyin.
3. **\*\*Reset (`RST`) Sonrası Davranış:\*\*** Cihaza `RST` komutu gönderildiğinde, mikrokontrolcü USB donanımını kapatıp yeniden başlatır. Bu durum bilgisayarda USB'nin sökülüp takılması etkisi yaratır. Programınızın bağlantıyı kesip, portun tekrar görünmesini beklemesi gerekir.