A blue and white logo with a circular design

Description automatically generated A red circle with a flower and a star

Description automatically generated

Rubik’s Cube Solver Robot

raspberry pi 2 ile destekleniyor.

**Ekip :**

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | 2020123142 | ZIEAD HASSAN | Python kodu, donanım  Ayarlama ve robot çalışması |
| **2** | 2022123164 | Salahaldeen Alshaikhkhalil | yazılım ve web sitesi |
| **3** | 2022123134 | Mohamed Bakir | PCB kartı |
| **4** | 2021123167 | Abdallah Elgendy |
| **5** | 2022123165 | Mohamed Aly | Raspberry Pi kurulumu |
| **6** | 2022123159 | Ahmed Algebaly | 3D baskı ve Robotu birleştiri |
| **7** | 2021123029 | ABDELAZIZ ELSAYED ALY |

**Introduction**

Bu proje, Raspberry Pi 2 kullanılarak tasarlanmış ve programlanmış bir Rubik Küpü çözme robotunu tanıtmaktadır. Robot, karmaşık algoritmalar ve görüntü işleme teknikleri sayesinde küpün durumunu analiz eder ve en hızlı çözüm yolunu hesaplar. Yenilikçi tasarımı ve teknolojik altyapısıyla bu robot, mühendislik ve yazılımın gücünü bir araya getirerek etkileyici bir performans sergilemektedir. Proje, hem algoritma geliştirme hem de robotik sistemlerin entegrasyonu açısından önemli bir öğrenme deneyimi sunmaktadır.

**Projenin github linki** : <https://github.com/Ziiadmaher/Rubik-s-Cube-solver>

**Projenin web sitesini linki** : https://github.com/SalahKhalill/Rubik\_cube

**projenin akış diagramı :**

**A diagram of a process

Description automatically generated**

**Bölümler:**

1. Malzemeler.

* Raspberry Pi 2
* Servo motorlar
* Kamera modülü (Raspberry Pi uyumlu)
* Sensörler (isteğe bağlı)
* LED ve dirençler
* Bağlantı kabloları ve jumper kablolar

1. PCB kartı.

* **Tasarım**
* Güç dağıtımı
* **Bağlantı Testi**

1. Raspberry Pi kurulumu.

* **Gerekli Yazılımları Yükleme**
* **Başlangıç Ayarları**
* **Robot Yazılımını Yükleme**

1. 3D baskı ve Robotu birleştiri.

* **3D Baskı Parçalar**
* **Birleştirme**

1. Ayarlama ve robot çalışması.

* **Kalibrasyon**
* **Test Aşaması**
* **Son Kontroller**

1. Malzemeler:

|  |  |  |  |
| --- | --- | --- | --- |
| adet | Malzeme | Fiyat (tl) |  |
| 2 | kajilai 2 adet DSSERVO DS3225 Dijital Servo 25 KG Metal Dişli Yüksek Tork | 1059 | A small red and black electronic device  Description automatically generated |
| 1 | Raspberry Pi Zero2W | 678 | A green circuit board with a black square and silver square and white square  Description automatically generated with medium confidence |
| 1 | microSDHC 16GB | 120 | A close-up of a memory card  Description automatically generated |
| 1 | PiCamera V1.3 | 213 | A green circuit board with a camera and white and blue ribbon  Description automatically generated |
| 1 | LED module 3W | 150 | A red and white electronic device  Description automatically generated |
| 1 | 2.5-5.5V TTP223  kapasitif dokunmatik sensör | 99 | A red circuit board with black and white text  Description automatically generated |
| 1 | USB Type-C BREAKOUT BOARD | 120 | A close-up of a usb port  Description automatically generated |

Elektrikli küçük parçalar:

|  |  |  |
| --- | --- | --- |
| adet | Parça | not |
| 1 | Prototype board | Tüm parçaları bağlamak için |
| 1 | 40pin (2x20) GPIO erkek başlığı | Raspberry Pi'nin WH sürümünü alamadıysanız |
| 1 | 40pin (2x20) GPIO dişi başlık  (Plastik gövde yüksekliği yaklaşık 8 ila 8,5 mm) | Bağlantı kartını Raspberry Pi Zero 2'ye bağlamak için |
| 4x3 | Erkek Başlıklar 90 derece | Servoları, dokunmatik sensör, LED modülünü bağlamak için |
| 2x3 | Kadın Başlıklar | Dokunmatik sensör bağlamak için, Led modülü |
| 3 | Kondansatör 16V 220uF | Servolar etkinleştirildiğinde voltaj düşüşünü sınırlamak için |
| 1 | Raspberry Pi için soğutucu |  |

Diğer:

Elbette bazı diğer yaygın malzemelere de ihtiyaç duyulur (teller, lehim ve lehim cihazı, lastik sargıları, kendinden yapışkanlı

kauçuk ayaklar, sıcak tutkal, vb).

Rubik Küpünü unutmayın

Küpün boyutu (kenar) 56,0 ile 57,5 ​​mm arasında olmalıdır.

1. Servos

A small red and black electronic device with a wire

Description automatically generated

* **Özellikler:**
  + Maksimum tork: 25 kg/cm
  + Çalışma voltajı: 6V-8.4V
  + Dişli malzemesi: Metal
  + Sinyal kontrol aralığı: PWM 500-2500 µs
  + Kullanım alanları: Yüksek tork gerektiren robotik sistemler ve model araçlar.
* **Avantajlar:**
  + Dayanıklı metal dişliler
  + Yüksek hassasiyet ve performans
  + Kolay montaj için standart boyutlar

1. Raspberry pi zero 2 w

A circuit board with many colors

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Pin | Label/GPIO | Amaç |
| 2, 4 | 5V | SBC kartını enerjilendirmak |
| 25,6 | GND | SBC kartını enerjilendirmak |
| 8 | GPIO 14 (UART TX | Rpi ON durumunun LED'i |
| 12 | GPIO 18 (PWM) | Top\_Cover PWM LED |
| 32 | GPIO 12 (PWM) | PWM Top\_Cover Servo |
| 33 | GPIO 13 (PWM) | PWM Cube\_holder Servo |
| 37 | GPIO 26 | Dokunmatik sensör |

1. Pi camera 1.3v

A close-up of a raspberry pi camera

Description automatically generated

**Teknik Özellikler:**

* Uyumlu Olduğu Raspberry Pi Modelleri: Raspberry Pi/3B/3B+/4/Zero
* Üretici Firma : China
* Boyutları : 25x20x9mm
* Çözünürlük : 5 MP (2592x1944 piksel)
* Video Çekimi : 1080p, 720p ve 640x480p
* Sensör Tipi : OmniVision OV5647 (5 megapiksel)
* Sensör Boyutu : 3.67 x 2.74 mmI
* Piksel Sayısı : 2592 x 1944
* Piksel Boyutu : 1.4 x 1.4 um
* Lens : f = 3,6 mm, f / 2,9
* Görüş Açısı : 54 x 41 derece
* Görüş Alanı : 2 m'de 2.0 x 1.33 m
* Tam kare SLR lens eşdeğeri : 35 mm
* Sabit Odak : Sonsuza 1 m
* Video : Codec H.264 (AVC) özellikli 30 fps'de 1080p
* Pano boyutu : 25 x 24 x 9mm (esnek kablo dahil değil)

2.PCB Kartı

A diagram of a machine

Description automatically generated

B) Prototip kart için notlar:

1. Prototip kart çift yüzlü tipte olmalıdır.

2. Kartı ve "sağ üst" köşeyi kesin.

3. 2x20 başlığını köşeden 3. deliğe yerleştirerek başlayın.

4. Diğer parçaların konumlandırılması için delikleri sayın.

5. Karşı taraftaki parçalar/teller için kesik çizgiler.

6. Mavi çizgiler yalıtımsız tellerdir.

7. Kırmızı, Kahverengi, Turuncu ve Yeşil çizgiler yalıtımlı tellerdir.

8. Diğer hatları geçerken yalıtımlı teller kullanın, ayrıca bunlar kartın karşı tarafında olduğunda.

9. Dolu noktalar bağlantılardır.

10. 3V3 voltaj regülatör kartı için maksimum yükseklik 15 mm'dir ve ekran

konnektörünün yakınındaki kapasitör.

12. Ø2,5 mm delik, ekranı desteklemek için bir M2,5 mm cıvata (ve 3 somun) içindir. En yüksek somun, kart yüzeyinden

~10,5 mm uzakta olmalıdır.

13. Kart test edildikten sonra servo konnektörlerine yakın son 2 kapasitörü ekleyin.

**Ön görünüş (2x20 başlık karşı tarafta):**

A diagram of a circuit board

Description automatically generated

**Arkadan görünüş :**

A circuit board with a number and a number

Description automatically generated with medium confidence

3. Raspberry Pi kurulumu.

**Adım 1:** Raspberry Pi Imager'ı şu adresten indirin: https://www.raspberrypi.com/software/

**Adım 2: İşletim sistemini microSD'ye aktarın:**

**a. CHOOSE DEVICE** seçin

**b. Raspberry pi zero 2 w** seçin

**A screenshot of a computer

Description automatically generated**

c. **Operating System** seçin

d. **Raspberry Pi OS (legacy, 32-bit) Full** seçin

A screenshot of a computer

Description automatically generated

e. STORAGE diskini seçin (microSD kart):

A screenshot of a computer

Description automatically generated

f. Ayarları kullanmak için YES seçin:

A screenshot of a computer

Description automatically generated

Raspberry Pi Imager, işletim sistemini microSD'ye yazmaya başlar ve bunu bir doğrulama takip eder.

Not: Yazma ve doğrulama yaklaşık 10 dakika sürer.

**Adım 3**: SD kartı Raspberry Pi'ye takın ve açın.

İlk başlatma iki dakikaya kadar daha uzun sürer; LED'in yanıp sönmesi durana kadar bekleyin.

1. 3D baskı

|  |  |  |  |
| --- | --- | --- | --- |
| Ref | Parça | Filaman (gram) | Baskı zamani |
| 1 | Structure | 145 | 14.5s |
| 2 | Top\_cover | 117 | 12.5s |
| 3 | Hinge | 51 | 5.5s |
| 4 | Baseplate front | 36 | 3.5s |
| 5 | Baseplate back | 37.3 | 3.5s |
| 6 | Cube\_holder | 34.4 | 3.5s |
| 7 | Cube\_lifter | 16.2 | 2s |
| 8 | Servo\_axis\_sup | 7.2 | 1s |
| 9 | Servo\_axis\_inf | 7 | 1s |
| 10 | PCB\_cover\_display | 41.4 | 4.5s |
| 11 | PiCamera holder | 6.5 | 0.5s |
| 12 | PiCamera holder frame | 18 | 2.5s |

Notlar:

1. En büyük kısmı Yapıdır ve 200x200mm plaka boyutuna sahip yazıcılara kolaylıkla sığar.

2. Filament uzunluğu Ø1,75 mm'ye dayanmaktadır.

3. Filament ağırlığı, PETG yoğunluğuna (1,23g/mm³) ve doğru sonuç için Ender 3 yazıcımda kullandığım yazdırma ayarlarına dayanmaktadır:

1. 0,2 mm katmanlar

2. Düşük hız (dış parçalar ve 1. katman için 25 ila 40 mm/s arası)

3. Dikey duvarlarda 4 katman

4. Yatay duvarlarda 5 katman

5. %30 dolum

6. 8mm kenar

4. Tablodaki filament miktarı ve baskı süresi üst sınır olarak dikkate alınmalıdır. Yazdırdıktan sonra

temel versiyon için toplam ağırlık, dilimleyici SW tarafından tahmin edilen 466 gramdan 400 grama daha yakındı.

5. ~ 8 saat gerektiren 3 (veya 4) parça daha yazdırarak Temel sürümü yükseltmek mümkündür.

6. Tüm parçalar çıkıntıları desteklemeden basılacak şekilde tasarlanmıştır.

7. Daha kolay ve daha iyi 3D baskı için bazı parçalar bölünmüştür.

8. 3D baskı için önerilen parça oryantasyonu aşağıdaki Tabloda gösterilmektedir.

|  |  |  |
| --- | --- | --- |
| Parça adı | Resim | 3D |
| Structure | A green metal piece with holes  Description automatically generated | A yellow plastic object with holes  Description automatically generated |
| Top\_cover |  |  |
| Hinge |  |  |
| Baseplate front |  |  |
| Baseplate rear |  |  |
| Cube\_holder |  |  |
| Cube\_lifter |  |  |
| PCB\_cover\_display |  |  |
| PiCamera holder |  |  |
| Servo\_axis\_sup |  | Symmetrical |
| Servo\_axis\_inf |  |  |
| PiCamera frame |  |  |

Robotun birleştirmesi

**Robotu monte etmeden önce:**

* Bağlantı panosunu yapın.
* Raspberry Pi'yi kurun.
* İki servo çıkış dişlisini orta konumlarına konumlandırın.

**Montaj sırası :**

1. Alt servoyu yapıya vidalayın
2. Servo\_axis\_inf / servo kolu / Servo\_axis\_inf sandviçini hazırlayın.
3. Cube\_holder'ı Servo\_axis düzeneğine birleştirin.
4. Cube\_holder düzeneğini alt servoya monte edin.
5. Menteşeyi Yapıya monte edin.
6. “T25” servo kolunu ancak servonun orta konumda olduğunu öğrendikten sonra üst servoya monte edin.
7. Top\_servo düzeneğini Top\_cover yuvasına yerleştirin.
8. Top\_cover düzeneğini Menteşeye monte edin.
9. Üst servo aksamını Menteşeye kadar tamamlayın.
10. Kaldırıcıyı Top\_cover'a monte edin.

**Adım 1:** (Alt servoyu yapıya monte edin):

4x M3x12mm silindirik kafa

Birkaç not:

* Dört vidayı sıkmadan önce servo çıkış dişlisinin Yapı deliğine iyi ortalanmış olup olmadığını kontrol edin.
* Aşağıdaki iki vidanın çıkıntısını sınırlamak için bu vidalara birkaç pul ekleyin; Bu, servo\_axis\_inf kısmına olası müdahaleyi önlemek içindir.

**Adım 2**: (Cube\_holder düzeneğini alt servoya monte edin):

Bu adım sırasında Servo çıkış dişlisini döndürmemeye çalışın: Diş bağlantısını yavaşça hissetmeye çalışın ve Cube\_holder iyi hizalanmamışsa geri çekin, Cube\_holder yaklaşık 90 derece döndürün ve tekrar kontrol edin.

Servo çıkışı ve Servo kolunun çift sayıda dişleri vardır.

**Adım 3:** (Menteşeyi Yapıya Birleştirin):

Not: Üç vida Yapının altından hafifçe çıkıntı yapacaktır; Vidalar 12 mm ise bu bir sorun olmayacaktır.

**Adım 4:** (Top\_servo düzeneğini Top\_cover yuvasına yerleştirin):

1x M3x12mm silindirik kafa

“T25” kolunun yuvası sıkı olabilir; Gerekirse fazla malzemeyi çıkarın.

M4 vida deliğinin vidayı kısıtlamadığından emin olun (Ø4,1 ila Ø4,3 mm olmalıdır).

**Adım 5:** (Top\_cover düzeneğini Menteşeye monte edin):

1x M4x20mm silindirik kafa

3x M3x12mm silindirik kafa

M4 vida deliğinin vidayı kısıtlamadığından emin olun (Ø4,1 ila Ø4,3 mm olmalıdır)

Üçüncü vida da yerleşene kadar iki M3x12mm'yi tamamen sıkmayın.

Not:



Servo çıkış dişlisi tarafında (yandaki ok) Top\_Cover ile Menteşe arasında boşluk olduğunu doğrulayın.

Çok az boşluk olması veya hiç boşluk olmaması durumunda, servonun M3 vidalarını sökün ve servo ile Menteşe arasına (tercihen vida konumlarına yakın) birkaç küçük ara parça (0,5 ila maksimum 1 mm) yerleştirin.

M3 vidalarını sıkın ve Top\_cover ile Menteşe arasındaki boşluğu yeniden kontrol edin.

**Adım 6:** (PiCamera tutucu çerçevesini Top\_cover'a monte edin):

4x M3x12mm silindirik kafa

PiCamera esnek kablosunu Top\_cover ile PiCamera tutucu çerçevesi arasına sıkıştırın.

**Adım 7:** (LED break board monte edin):

2x M3x12mm silindirik kafa ekleyin.

Alanın kısıtlı olması nedeniyle karttaki konnektör yerine kabloları çözmek daha uygun olacaktır.

Vida ucu panele temas ettiğinde vidalamayı bırakın.

**Adım 8:** (PiCamera esnek kablosunu Raspberry Pi Zero2 kartına bağlayın ve Yapıya sabitleyin)

4x M2,5x4mm (veya M2,5x maksimum 12 mm'ye kadar) silindirik kafa.

**Adım 9:** (PCB\_cover'ı birleştirin):

2x M3x12mm konik kafa

Kabloların Yapıya doğru sıkışmamasına dikkat edin.

**Adım 10:** (PiCamera modülünü PiCamera tutucuya monte edin):

Ø1,75 mm'lik küçük filament parçaları gereklidir.

4 parçayı tutucuya bastırın, tahtayı kaydırın, sıcak bıçakla filamanı deforme edin.

**Robotu birleştirdikten sonra şöyle görünecek:**

A white plastic device with a lid open

Description automatically generatedA white and blue cube with wires

Description automatically generated

5. Ayarlama ve Kodlar.

**Servo açılarının ayarlanması:**

Sarf malzemeleri listesindeki servolar 180° dönüşe sahiptir; bu, bu robotun kaldırıcı ve Üst\_kapak açısı için fazlasıyla yeterlidir ve Cube\_holder için yeterince doğru olmalıdır; Açı çözünürlüğünü etkileyeceği için 270° servo almanızı önermiyorum.

Farklı servolar arasındaki toleransların yanı sıra, bir varyasyon kaynağı da servo kolları ile servonun çıkış dişlisi arasındaki bağlantılardır ve birçok olası konuma sahiptir (25 diş olduğuna inanıyorum).

Bu, robotumda düzgün çalışan Cubotino\_T\_servo\_settings.txt dosyasında ayarlanan referans açılarının diğer sistemlerde mutlaka en iyi seçim olmadığı anlamına gelir: Bu parametreler her sistemde ayarlanmalıdır!

Servolar bir PWM sinyali aracılığıyla açılı olarak kontrol edilir (https://en.wikipedia.org/wiki/Servo\_control)

Sarf malzemeleri listesindeki servolar, 1 ms'den 2 ms'ye kadar bir Darbe Genişliği sinyalini kabul eder; burada 1,5 ms, orta açıdır.

Farklı darbe genişliği aralığına sahip servolar kullanmak yine de mümkündür; bu durumda ilgili servo için min\_pulse\_width ve max\_pulse\_width parametrelerini ayarlayın, daha fazla bilgi için bkz. Parametreler ve ayarlar.

Cubotino\_T\_servos.py, servo PWM'yi yönetmek için gpiozero kütüphanesini kullanır.

Bu kitaplık, Darbe Genişliği aralığına (örn. 1 ila 2 ms veya 500 ila 2500us) bağlı olarak -1 ila 1 (0 orta açıdır, değer kayan noktadır) arasında değişen bir parametreye sahip bir hedef servo konumu/açı kullanır. )

**Top\_cover (t\_servo)** açıları:

Servoların çalışma açıları bir (Json) metin dosyasında ayarlanır: Cubotino\_T\_servo\_settings.txt.

Tanımlanmış 5 açı vardır (çoğunlukla konum olarak bahsedilir):

* Close: üst ve orta küp katmanlarını sınırlandıracak konum.
* Rel: Kapalı konumda küpteki gerilimi serbest bırakma konumu.
* Open: Küp ve Küp\_tutucusu ile çakışmayan konum.
* Read: Kaldırıcı neredeyse küpe değecek şekilde kamera okuması konumu

(ve ne yazık ki Cube\_holder'ı kısıtlıyor).

* Flip: Kaldırıcının küpü çevireceği konum (yaklaşık 2 küp katmanı yüksekliğinde).

**Cube\_holder (b\_servo)** açıları:

|  |  |
| --- | --- |
| There are 7 defined angles | **Description** |
| Home | CCW ve CW arasındaki orta konum |
| CCW | Cube\_holder'ı Ana Sayfadan yaklaşık 90° CCW yönünde döndürme veya döndürme konumu  (Motorun bakış açısına göre yön) |
| CW | Cube\_holder'ı Başlangıç ​​konumundan yaklaşık 90° saat yönünde döndürme veya döndürme konumu  (Motorun bakış açısına göre yön) |
| Release\_CW | CW'de küp gerilimini serbest bırakın (daha önce b\_extra side olarak adlandırılıyordu) |
| Release\_CCW | CCW'de küp gerilimini serbest bırakın (daha önce b\_extra side olarak adlandırılıyordu) |
| Extra\_home\_CCW | CCW'den dönerken evde küp gerginliğini serbest bırakın (daha önce b\_extra home olarak adlandırılıyordu) |
| Extra\_home\_CW | release cube tension at home, when rotating from CW (called *b\_extra home* before) |

**Kod Yapısı:**

**1. Gerekli Kütüphaneler:**

Kod, çeşitli işlemler için birçok kütüphane kullanır:

* **argparse**: Komut satırı argümanlarını işlemek için.
* cv2 (OpenCV): Görüntü işleme ve analiz için.
* RPi.GPIO: Raspberry Pi GPIO pinlerini kontrol etmek için.
* numpy: Matematiksel işlemler ve veri analizi için.

**2. Komut Satırı Argümanları**

Kod, komut satırından alınan argümanlarla çalıştırılabilir. Örneğin:

* **--debug**: Hata ayıklama modunu etkinleştirir.
* **--cycles:** Otomatik çözüm döngülerinin sayısını belirler.

**Fonksiyonlar**

Kodun büyük bir kısmı, modüler bir yaklaşımla yazılmış fonksiyonlardan oluşur. Aşağıda her bir fonksiyonun detaylı açıklaması verilmiştir:

**1. import\_parameters(debug=False)**

**Açıklama**: JSON formatında bir ayar dosyasından robotun çalışma parametrelerini alır. Bu, robotun farklı yapılandırmalar için kolayca ayarlanmasını sağlar.

**Girdi**:

* debug (bool): Hata ayıklama çıktısını etkinleştirmek için.

**Çıktı**:

* Başarılıysa parametreler yüklenir, aksi halde hata döndürür.

**2. read\_facelets(frame, w, h)**

Açıklama: Kameradan alınan görüntüyü işler ve küp yüzeyindeki kareleri algılar. OpenCV'nin kenar algılama ve kontur bulma yöntemlerini kullanır.

Girdi:

* frame: Kameradan alınan bir görüntü.
* w, h: Görüntü genişliği ve yüksekliği.

Çıktı:

* Algılanan konturlar ve kontur hiyerarşisi.

**3. Robot\_camera\_setting(debug, os\_version, camera, face)**

Açıklama: Kamera ayarlarını stabilize eder ve tutarlı görüntüler elde etmek için gerekli değişiklikleri yapar.

Girdi:

* debug (bool): Hata ayıklama modu.
* os\_version: Raspberry Pi OS sürümü.
* camera: Kamera nesnesi.
* face: Küpün analiz edilen yüzü.

Çıktı:

* Stabilize edilmiş kamera parametreleri.

**Kamera İşleme**

Kod, küp durumunu algılamak için kamera görüntülerini işler. Aşağıdaki işlemler gerçekleştirilir:

1. Görüntü Kırpma: Fazlalık alanlar kaldırılır.
2. Perspektif Düzeltme: Görüntü, küpün üstten görünümüne benzer hale getirilir.
3. Kenar Algılama: Küp yüzeyindeki karelerin sınırları belirlenir.

**Kociemba Algoritması Nedir?**

Kociemba algoritması, Rubik Küpü çözmek için kullanılan optimize bir yöntemdir ve iki aşamada çalışır:

1. Faz 1 (Phase 1): Küp, belirli bir ara duruma (örneğin, tüm kenarların hizalanmış olduğu bir duruma) getirilir.
2. Faz 2 (Phase 2): Küp, ara durumdan tamamen çözülmüş duruma getirilir.

**Kociemba Algoritması Kullanımının Adımları**

1. Küp Durumunu Algılama:
   * Robot, PiCamera ve OpenCV kullanarak küpün renklerini tanır.
   * Her yüzeyin rengi U, F, R, B, L, D ile temsil edilir (üst, ön, sağ, arka, sol, alt).
2. Durum Dizisini Hazırlama:
   * Tüm yüzlerin renkleri tek bir diziye dönüştürülür.
3. Kociemba Çözücüsünü Kullanma:
   * Kociemba çözücüsü, durumu analiz ederek çözüm adımlarını hesaplar.
4. Hamleleri Uygulama:
   * Çözüm adımları servo motorlara aktarılır ve küp çözülür.