

DENEYAP

Teknoloji Atölyeleri

**ROBOTİK VE
KODLAMA**

ORTAOKUL

Dr. Öğr. Üyesi Memet ÜÇGÜL

Prof. Dr. İbrahim ÇETİN

Prof. Dr. Erman YÜKSELTÜRK

Prof. Dr. Ercan TOP



TÜBİTAK Deneyap Kitapları 3

ROBOTİK VE KODLAMA
ORTAOKUL

Dr. Öğr. Üyesi Memet ÜÇGÜL
Prof. Dr. İbrahim ÇETİN
Prof. Dr. Erman YÜKSELTÜRK
Prof. Dr. Ercan TOP

© Türkiye Bilimsel ve Teknolojik Araştırma Kurumu, 2021

Bu kitabın bütün hakları saklıdır.
Yazılar ve görsel materyaller, TÜBİTAK'tan yazılı izin alınmadan
tümüyle veya kısmen çoğaltılamaz ve yayımlanamaz.
Kitabın PDF formatındaki elektronik nüshasına
<https://yayinlar.tubitak.gov.tr/deneyap-atolyesi> adresinden ulaşılabilir.
TÜBİTAK Deneyap Kitapları DENEYAP TÜRKİYE Projesi kapsamında hazırlanmıştır.

ISBN 978-605-312-438-2
Yayıncı Sertifika No: 47703

Yayın Tarihi: 2023

TÜBİTAK Başkanı: Prof. Dr. Hasan MANDAL
Bilim ve Toplum Başkanı: Ömer KÖKÇAM
Genel Yayın Yönetmeni: Fatma BAŞAR
Editörler: Kübra BAL ÇETİNKAYA, İpek PİRPIROĞLU GENCER
Düzeltili: Özlem KÖROĞLU, Adem ULUDAĞ, Şermin ASLAN
Telif İşleri Sorumlusu: Havva Hilal KAÇAR

TÜBİTAK Bilim ve Toplum Başkanlığı
Tunus Caddesi No: 80 Kavaklıdere 06680 Ankara
Tel: (312) 298 96 50
e-posta: deneyap@tubitak.gov.tr
<https://yayinlar.tubitak.gov.tr/deneyap-atolyesi>

İçindekiler

ÖĞRETİM KILAVUZU	1
ROBOTİK KODLAMA DERSİ ÖĞRETİM KILAVUZU.....	1
GÖZLE - UYGULA - TASARLA - ÜRET - DEĞERLENDİR ÖĞRENME DÖNGÜSÜ.....	3
EŞLİ PROGRAMLAMA.....	5
GRUPLAR ARASI İLETİŞİM VE ROBOT YARIŞLARI	6
ROBOTİK KİTLER VE PROGRAMLAMA ORTAMI.....	6
Lego Mindstorms Education EV3	6
EV3 Yazılımı	7
MicroPython.....	8
Eğitimde Kullanılacak Mat (Çalışma Alanı).....	8
Kaynakça	10
1. HAFTA: ROBOT VE ALGORİTMA	11
1. GÖZLE VE UYGULA	12
1.1. Gözle: Robot Kavramı Üzerine Tartışma	12
1.2. Uygula: Algoritma Öğreniyorum.....	13
1.3. Uygula: Çarpım Eşleri Algoritması	13
2. TASARLA	14
2.1. “Hayalimdeki Robot” Etkinliği.....	14
3. ÜRET.....	14
3.1. “İlk Robotumu Yapıyorum” Etkinliği.....	14
4. DEĞERLENDİR.....	15
5. İLAVE ETKİNLİK.....	16
5.1. Algoritma oyunu (Sırt sırta oturup ayakkabı bağcığı bağlama).....	16
2. HAFTA: ROBOTLAR İLE HAREKET	18
1. GÖZLE VE UYGULA	19
1.1. Gözle: EV3 Yazılımı	19
1.2. Gözle: Programlama Alanının Kullanılması	22
1.3. Gözle: Robotun Hareket Ettirilmesi.....	23
1.4. Hareket Blokları.....	24
1.4.1. Gözle: Direksiyon Hareketi (Move Steering).....	24
1.4.2. Uygula: Tekerin Yarıçapını Hesaplama	25
1.4.3. Gözle: Palet Hareketi (Move Tank)	25
1.4.4. Uygula: Palet Hareketi	26
1.4.5. Gözle: Büyük Motor (Large) ve Orta (Medium) Motor Blokları	26
1.4.6. Uygula: Orta Motor.....	27
2. TASARLA	28
3. ÜRET.....	29
4. DEĞERLENDİR.....	29
5. İLAVE ETKİNLİK.....	30

5.1. Sonsuz İşareti Şeklinde Hareket Etme.....	30
5.2. Yarışma.....	30
3. HAFTA: ROBOTLARLA SES, METİN, RESİM	31
1. GÖZLE VE UYGULA	32
1.1. Gözle: Ses (Sound) Bloğu	32
1.2. Uygula: Ses (Sound) Bloğu	34
1.3. Gözle: Siren Sesi.....	34
1.4. Uygula: Kare Çizen Robot	36
1.5. Gözle: Ekran (Display) Bloğu.....	37
1.5.1. Gözle: Metin (Text)	39
1.5.2. Gözle: Resim (Image) Dosyaları	39
1.5.2. Gözle: TÜBİTAK Logosunu Yükleme	40
1.6. Uygula: Ekran (Display) Bloğu	41
1.6.1. Metin Yazdırma.....	41
1.6.2. Resim (Image) Basma	42
1.6.3. Şekil Oluşturma.....	42
1.7. Gözle: Tuğla Durum Işığı (Brick Status Light) Bloğu	42
1.8. Uygula: Tuğla Durum Işığı (Brick Status Light) Bloğu	43
2. TASARLA	43
2.1. Dans Eden Robot	43
3. ÜRET	44
3.1. Dans Eden Robot	44
4. DEĞERLENDİR	45
5. İLAVE ETKİNLİK.....	47
5.1. Dans Eden Işıklı Robot.....	47
4. HAFTA: DOKUNMA SENSÖRÜ	48
1. GÖZLE VE UYGULA	49
1.1. Gözle: Sensör Nedir?	49
1.2. Uygula: Engele Çarpınca Geri Dönen Robot	52
1.3. Uygula: Engellere Çarpan ve Her Seferinde Geri Dönme İşlemini Yapan Robot.....	53
1.4. Gözle: Anahtar (Switch) Bloğu.....	54
1.5. Uygula: Koşul Durumunu Göster	55
2. TASARLA	56
2.1. Oyuncak Köpeğe Çarpınca Havlayan Robot	56
3. ÜRET	56
4. DEĞERLENDİR	57
5. İLAVE ETKİNLİK.....	57
5.1. Çift Sensör Kullanımı	57
5. HAFTA: MESAFE SENSÖRÜ	59
1. GÖZLE ve UYGULA	60
1.1. Gözle: Mesafe Sensörü Nedir, Nasıl Kullanılır?	60
1.2. Gözle: Belirli Bir Mesafeye Kadar İlerleme.....	61
1.2. Uygula: İstenilen Mesafe Kadar Geri Gitme	63
1.3. Gözle: Tekerin Yarıçapını Hesaplama	63

1.4. Uygula: Tekerin Yarıçapını Hesaplama	64
1.5. Uygula: Engele Yaklaştıkça Yavaşlayan Robot	64
2. TASARLA VE ÜRET	66
2.1. Birinci Görev: Öndeki Aracı Takip Eden Robot	66
2.2. İkinci Görev-Takımlar Yarışıyor	67
3. DEĞERLENDİR	67
4. İLAVE AKTİVİTE	68
4.1. Park Sensörü.....	68
6. HAFTA: AÇI SENSÖRÜ	69
1. GÖZLE VE UYGULA	70
1.1. Gözle: Açı Sensörü.....	70
1.2. Uygula: 360 Derece Dönen Robot.....	71
1.3. Gözle: Açı Sensörü Kalibrasyonu	72
1.4. Uygula: Açı Sensörü Değerini Ekranaya Yazma.....	73
1.5. Uygula: Açı Sensörünü Kullanarak Kare Şeklinde Hareket Eden Robot	73
2. TASARLA	73
2.1. Açı Sensörünü Kullanarak Robotun Düz Gitmesini Sağlamak	73
3. ÜRET	74
4. DEĞERLENDİR	74
5. İLAVE ETKİNLİK.....	75
5.1. Kalemle Zemine Çokgen Çizdirme	75
5.2. Kalemle Zemine İstenilen Çokgeni Çizdirme	76
7. HAFTA: RENK SENSÖRÜ	77
1. GÖZLE VE UYGULA	79
1.1. Gözle: Renk Sensörü Nedir ve Nasıl Kullanılır?.....	79
1.2. Gözle ve Uygula: Karşılaşılan Rengin İngilizcesini Söyleyen Program.....	79
1.3. Gözle: Yansıyan Işık Şiddeti.....	81
1.4. Uygula: Dairenin İçerisinden Çıkmayan Robot.....	83
1.5. Gözle: Ortam Işığı	86
1.6. Uygula: Ortam Işığı.....	86
2. TASARLA VE ÜRET	88
2.1. Tasarla: Eliptik Bir Yörüngeyi Takip Eden Robot	88
2.2. Üret: Eliptik Bir Yörüngeyi Takip Eden Robot.....	89
2.3. Tasarla: Dikdörtgen Üzerinde Hareket Eden Robot	90
2.4. Üret: Dikdörtgen Üzerinde Hareket Eden Robot	91
2.5. Tasarla: Üçüncü Çizgiyi Geçince Duran Robot.....	91
2.6. Üret: Üçüncü Çizgiyi Geçince Duran Robot	92
3. DEĞERLENDİR	92
8. HAFTA: SENSÖRLERİN BİRLİKTE KULLANIMI.....	94
1. GÖZLE VE UYGULA	95
1.1. Gözle: Farklı sensörleri bir arada kullanma.....	95
1.2. Uygula: Farklı sensörleri bir arada kullanma.....	96
1.3. Gözle: Mantık İşlemi (Logic Operation) Bloğu.....	97
1.4. Uygula: OR (Veya) Mantık İşlemi.....	99

1.5. Uygula: AND (Ve) Mantık İşlemi	99
2. TASARLA VE ÜRET	99
2.1. Tasarla: Çizgi İzleyen ve Engelde Duran Robot.....	99
2.2. Üret: Çizgi İzleyen ve Engelde Duran Robot.....	100
2.3. Tasarla: Çizgi İzleyen ve Engeli Aşan Robot	101
2.4. Üret: Çizgi İzleyen ve Engeli Aşan Robot	101
3. DEĞERLENDİR.....	101
4. İLAVE ETKİNLİK.....	102
4.1. Nesnelere Arasından Belirli Renklerdeki Toplamak	102
9. HAFTA: DEĞİŞKENLER	103
1.1. Gözle: Değişkenler	105
1.2. Uygula: Grup Sorusu	105
1.3. Gözle: EV3 Yazılımında Değişken Tanımlama	105
1.4. Gözle ve Uygula: EV3 Yazılımında Değişken Kullanma	107
1.5. Gözle ve Uygula: Karşısına Engel Çıktığında Sağından Geçen Robot	109
2. TASARLA VE ÜRET	111
2.1. Tasarla: Çizgi Takip Ederken Engel Aşan Robot	111
2.2. Üret: Çizgi Takip Ederken Engel Aşan Robot	111
3. DEĞERLENDİR.....	112
5. İLAVE ETKİNLİK.....	112
5.1. Tasarla: Şifreli Sayıyı Bulma Yarışması	112
5.2. Üret: Şifreli Sayıyı Bulma Yarışması	114
5.3. Tasarla: Renk Sayacı	114
5.4. Üret: Renk Sayacı	115
10. HAFTA: PARALEL İŞLEMLER	116
1. GÖZLE VE UYGULA	117
1.1. Gözle: Paralel İşlemler	117
1.2. Uygula: Paralel İşlemler.....	119
1.3. Gözle: Deney Arayüzü ve Veri Kaydı.....	119
1.4. Uygula: Motor Hareket Verisinin İncelenmesi	121
1.5. Uygula: Motor ve Mesafe Sensörü Verilerinin Karşılaştırılması.....	122
2. TASARLA	124
2.1. Tasarla: Renk Tahmini Oyunu	124
3. ÜRET.....	126
3.1. Motor ve Açık Sensör Verilerini Kullanarak Rota Oluşturma	126
4. DEĞERLENDİR.....	127
5. İLAVE ETKİNLİK.....	128
5.1. Diğer Grubun Hareketini Tahmin Edip Uygulanan Programı Oluşturma.....	128
11. HAFTA: KENDİ BLOĞUNU OLUŞTURMA.....	129
1. GÖZLE	130
1.1. Büyük Problemler Küçük Problemlerden Oluşabilir.....	130
1.2. Labirent Çözme	130
1.3. Görevler	131
1.4. Robot Gereksinimleri	133

1.5. Alt Problemler	133
2. UYGULA	134
2.1. Duvar Takip	134
2.2. Sola Dönme	136
2.3. Sağa Dönme	137
2.4. Görevlerin Birleştirilmesi	138
3. TASARLA VE ÜRET	139
3.1. Labirent Yarışması	139
3.2. Tasarla	140
3.3. Üret	140
3.4. Labirent Çözümünün İyileştirilmesi	140
3.5. Labirent Görevinin Zorlaştırılması	141
4. DEĞERLENDİR	141
12. HAFTA: BLUETOOTH	142
1. GÖZLE VE UYGULA	143
1.1. Gözle: Bluetooth	143
1.2. Gözle: Bluetooth Bağlantısı ve Mesajlaşma Blokları	146
1.3. Uygula: Merhaba Mesajı	147
1.4. Uygula: Sensör Verilerinin Diğer Robota Aktarılması	148
2. TASARLA	149
2.1. Tasarla: Taklitçi Robot	149
3. ÜRET	150
4. DEĞERLENDİR	152
EK - PROJE HAZIRLIYORUM	154
1. HAZIRLIK (Projeye Başlama)	154
2. EMPATİ	155
3. TANIMLAMA	155
4. FİKİR ÜRETME	156
5. GELİŞTİRME – TEST ETME	157
6. SUNUM	157

Sunuş

İçinde bulunduğumuz çağda öğrenci rolleri değişmiştir. Öğrencilerden öğrendiği kavram, prosedür ve temel bilgileri başka bağlamlara aktarması, bu bilgileri kendi amaçları doğrultusunda yeniden inşa etmesi ve yeni bilgiler oluşturması beklenmektedir. Bu amaç doğrultusunda, bu kitapta öğrencilere bilgi işlemsel düşünme ve problem çözme gibi üst düzey düşünme becerileri kazandırmaya yönelik etkinlikler sunulmuştur. Günümüzde bilişim teknolojileri okur-yazarlığı sadece teknolojik araçları kullanmak olarak ifade edilmemektedir. Bilişim teknolojileri okur-yazarlığı daha kapsamlı olarak düşünülmektedir ve artık bu araçların üretimini de içermektedir. Birçok araştırmacı herkesin bilgisayar biliminin temel kavramlarını bilip kullanması gerektiğini söylemiştir. Bu bağlamda bilgi işlemsel düşünme 21. yüzyılda yaşayan bütün bireyler için temel bir beceri olarak kabul edilmektedir.

Robotlar ve robotik programlama, öğrencilerin bilgi işlemsel düşünme becerisini geliştirmek için uygun bir ortam sunmaktadır. Bu ortam uygun bir pedagoji ile birleştirildiğinde öğrencilerin problem çözme, yaratıcılık ve bilgi işlemsel düşünme gibi üst düzey bilişsel becerilerini geliştirmesine yardımcı olabilir. Bu kitapta öğretmenlere, öğrencilerin bahsi geçen becerilerini robotik programlama vasıtasıyla geliştirmeleri sürecinde rehberlik etmek hedeflenmektedir. Öğretmenlerin öğrencilerin gelişmesine yardımcı olabilmesi için bir öğretim modeli kullanması faydalı olabilir. Bu amaç doğrultusunda bu kitabın yazarları tarafından “Gözle, Uygula, Tasarla, Üret ve Değerlendir” öğrenme döngüsü geliştirilmiştir. GUTÜD öğrenme döngüsünde temel kavramların oluşturulmasında etkinlik ve öğretmen temelli göster ve uygula yaklaşımı ön plana çıkarılırken, öğrencilerin verilen temel bilgiler ile ileri seviye zihinsel faaliyetlerde bulunması için keşfe dayalı yaklaşımlar ön plana çıkarılmaya çalışılmıştır. Gözle bölümünde öğretmen bir robotik konusunu uygulamalı olarak (göstererek) anlatır. Uygula bölümünde öğretmen öğrencilerden Gözle bölümünde gösterilen uygulamaların aynısını / bir benzerini ister veya onlarla birlikte yapar. Tasarla bölümünde öğrencilerden kendilerine verilen karmaşık bir problemin çözümünü tasarlamaları istenir. Üret bölümünde öğrencilerden tasarladığı planları kullanarak problem için algoritmik bir çözüm üretmesi istenir. Değerlendir bölümünde hedef, öğrencinin, öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu öğretim döngüsünde öğrencilerin kendilerine verilen görevlerden uygun olanlarını eşli programlama grupları içerisinde yapmaları önerilmiştir. Eşli programlamada, iki öğrenci bir robot veya bilgisayar karşısında yan yana oturarak tasarım, algoritma, kod yazma veya hata ayıklama için iş birlikli çalışır.

Kitapta robot olarak LEGO firması tarafından üretilen Mindstorms Education EV3 seti kullanılmıştır. Bu setle oluşturulan robotları programlamak için blok tabanlı programlama ortamı olan EV3 yazılımı kullanılmıştır. Ayrıca etkinlikler boyunca kullanılmak üzere 70 cm x 100 cm boyutunda çift taraflı MAT (Çalışma Alanını) tasarlanmıştır. Etkinliklerin yapılabilmesi için bu MAT'ın kullanılması gerekmektedir. Elinde MAT bulunmayan öğretmenler her etkinlik için MAT'taki şekillerin bir benzerini oluşturarak kullanabilir.

Bu kitap ortaokul seviyesinde robotik kodlama eğitimi vermek isteyen öğretmenler için hazırlanmıştır. Kitap içerisinde basitten karmaşığa gidecek şekilde spiral bir yaklaşım kullanılarak konular verilmiştir. Konuların tamamı GUTÜD döngüsü çerçevesinde oluşturulmuştur. Her ne kadar kitaptaki amaç bu olmasa da, kodlama hakkında ön bilgisi olmayan öğretmenler de konuyu öğrenmek için bu kitabı kullanabilir. Bunun

yanında ileri seviye programlamaya ve robotik kavramlarına giriş yapmak için kitabın lise seviyesi için hazırlanan basımında kullanabilir.

Kitap 12 haftalık bir ders düşünülerek planlanmıştır. Her haftaya bir bölüm düşecek şekilde on iki haftalık içerik hazırlanmıştır. Öğrenciler 8. haftadan itibaren her hafta proje konuları ile ilgili de yönlendirilebilir. Gerek duyulması durumunda projeler hazırlamak için ek sürelerde ayrılabilir. Bu sırayı izlemek ve kitabın tamamını kullanmak istemeyen öğretmenler kitap içerisindeki ilgili bölümleri kendi dersleri için kullanabilirler. Fakat bu kitapta konular bütünlük teşkil edecek şekilde tasarlanmıştır.

Sevgili öğretmenler kitabın size ve ülkemiz öğrencilerine faydalı olması dileğiyle!

Öğretim Kılavuzu

ROBOTİK KODLAMA DERSİ ÖĞRETİM KILAVUZU

İçinde bulunduğumuz çağda öğrenme, öğrenenlerin sınıf içinde verilen kavram, prosedür ve temel bilgileri istenildiği zaman belirli bir ölçüde, sınav veya başka vasıtalar ile yeniden göstermesi olarak algılanmamaktadır. Amerikan Ulusal Araştırma Konseyi (2012), 21. yüzyıl öğrenenleri için bilişsel, kişisel, kişiler arası alan olmak üzere üç temel yeterlik başlığı belirlemiştir. Öğrenenlerin öğrendiği kavram, prosedür ve temel bilgileri başka bağlamlara aktarması ve bu bilgileri kendi amaçları doğrultusunda yeniden inşa etmesi gerekmektedir. Bu amaç doğrultusunda, öğrenenler problem çözme gibi üst düzey düşünme becerileri kazandırmaya yönelik etkinlikler sunulmalıdır.

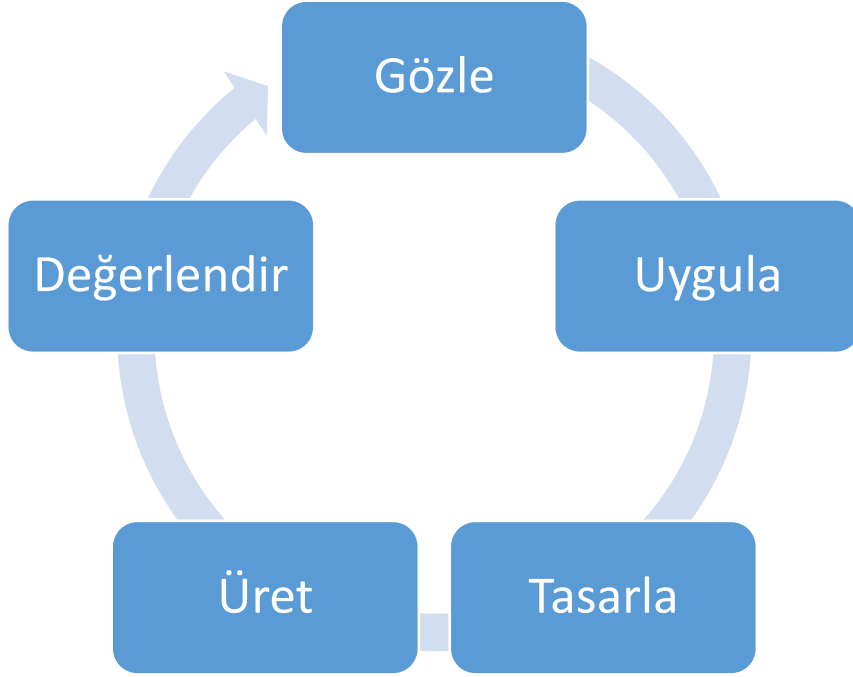
Problem çözme doğal, karmaşık ve anlamlı bir öğrenme/düşünme aktivitesi olarak tanımlanmaktadır. Problemlerin; otantik, öğrenenlerin yaşantılarıyla ilişkili, derin öğrenmeyi destekleyen öğeleri içermiş ve içerdiği bilgi ve becerilerin zorluğunun hiyerarşik olarak yapılandırılmış olması gerekmektedir (Jonassen, 2007). Öğrencileri gerçekçi ve bütünsel görevlere dâhil etmenin, onların uygun şema ve zihinsel modeller oluşturmasına yardımcı olacağına inanılmaktadır (Merrill, 2007). Problemler doğaları gereği yapısalılık, karmaşıklık, dinamiklik ve meydana geldikleri bağlam açısından farklılıklar göstermektedir. Bu yüzden iyi oluşturulmuş içsel temsillerin, öğrenenlerin yeni edindikleri bilgi ve becerileri daha sonra farklı durumlarda uygulamalarını kolaylaştırdığına inanılmaktadır. Öğrenenler meslek yaşamlarında, daha önce hiç karşılaşmadıkları, farklı ve ileri seviye problemler ile karşılaşacakları için yaratıcı düşünmek durumundadırlar. Öğrenenler aynı zamanda nasıl öğrenmesi ve kendine nasıl yön çizmesi gerektiğini de öğrenmelidirler. Tüm bunlara ek olarak öğrenenler, iş birliğine dayalı çalışma ortamlarında nasıl çalışacaklarını ve nasıl iletişim kurmaları gerektiğini bilmelidirler.

Yaşadığımız çağda bilgi teknolojilerindeki değişimler ve bunun topluma ve günlük yaşama yansımaları, mesleklerdeki değişimler ve fen bilimleri ile matematik alanlarındaki teknoloji kullanımı bireylerin bilişim teknolojileri okur-yazarı olmasını gerektirmektedir. Fakat bilişim teknolojileri okur-yazarlığı, sadece teknolojik araçları kullanmaktan ibaret değildir. Bu araçların üretimini de içermektedir. Wing (2006), herkesin bilgisayar biliminin temel kavramlarını bilip kullanması gerektiğini söylemiştir. Wing'e göre bilgi işlemsel düşünme, 21. yüzyılda yaşayan bütün bireyler için temel bir beceridir. Cuny, Snider ve Wing (2010) bilgi işlemsel düşünmeyi "Çözümlerin bir bilgi işleme birimi tarafından etkili şekilde yerine getirilebilecek formda sunulması amacıyla problemleri ve çözümleri formülleştirmeyi içeren düşünme süreci" olarak tanımlamışlardır. Bilgi işlemsel düşünme bir problem çözme sürecidir ve beş bileşeni içerir: (i) soyutlama, (ii) algoritmik düşünme, (iii) problemi bileşenlerine ayırma, (iv) çözümü verim açısından analiz etme ve (v) bir çözümü farklı bağlamlar için genelleme.

Bilgi işlemsel düşünme terimini ilk olarak kullanan ve programlamanın bütün öğrenciler tarafından öğrenilmesi gerektiğini bir teorik çerçeve ile ortaya koyan ilk kişi Seymour Papert'tir (Papert, 1991). Papert, Piaget ile çalışmış ve onun yapılandırmacılık (constructivism) teorisinden etkilenmiştir. Papert (1993) yapılandırmacılığı temel alarak onun bir yorumu olan inşacılık (constructionism) yaklaşımını geliştirmiştir. Yapılandırmacılık, pedagojik anlamda basit bir şekilde ele alınacak olursa, öğrencinin öğrenme sürecindeki rolünü, pasif olarak bilgiyi alandan, aktif bir şekilde bilgiyi inşa eden olarak değiştirir. Yapılandırmacılık, felsefi anlamda basit bir şekilde ele alınacak olursa, dışsal objektif bilginin varlığını kabul etmez. Böyle bir bilgi olsa bile dışsal 'gerçeklik' öğrenen tarafından aynen kavranamaz. Öğrenen her hâlükârda kendi bilgisini inşa etmelidir. Yapılandırmacılık, öğretmeni bilgiyi bir zihinden diğerlerinin (öğrenenlerin) zihnine aktaran pedagojik bir araç olarak görmez. Öğretmen öğrenene buluş, keşif, inşa veya yeniden inşa sürecinde yardımcı olur ve onu pedagojik olarak destekler. Papert'ın inşacılığı, yapılandırmacılığın bu yorumlarını aynen alır ve bilginin inşa sürecinde öğrenenlerin somut bir ürün vasıtası ile soyut kavramsal bilgiye ulaşması gerektiğini savunur. Öğrenenler zihinlerindeki temeli kullanarak somut bir ürün geliştirirler. Bu ürünü geliştirirken soyut kavramları kullanmaları gerekir. Böylece soyut kavramlar somut ürünler vasıtası ile öğrenenin öğrenme alanına dâhil olur. Öğrenenin kendisi, diğer öğrenenler ve öğretmen bu somut ürün hakkında konuşarak öğrenenin soyut bilgiyi içselleştirilmesine yardımcı olabilir. Böylece öğrenenler sosyal bir ortamda somut ürünler vasıtası ile bilgilerini inşa ederler.

Bilgi işlemsel düşünmenin geliştirilmesi için çeşitli yaklaşımlar kullanılabilir. Bu yaklaşımlara örnek olarak; hiçbir bilgisayar gerektirmeyen bilgisayarsız bilgisayar bilimi, görsel programlama, metin tabanlı programlama ve robotik programlama verilebilir. Bu dersin içeriğini robotik programlama oluşturmaktadır. Bu derste öğrenenlerin bilgi işlemsel düşünme becerilerini robotik programlama vasıtasıyla geliştirmelerine yardımcı olmak hedeflenmektedir. Programlama başlı başına zor bir kavramdır ve çeşitli zihinsel zorluklar içerir (Çetin, 2013). Burada bu zorluklara değinilmeyecektir fakat bahsi geçen zorluklardan dolayı programlama öğretilirken temel pedagojik çerçevenin dikkatlice belirlenmesi gerekir. Bu dersin tasarımında temel olarak 21. yüzyıl yeterlikleri ve programlama eğitiminin alan yazını kullanılacaktır. Programlama eğitimiyle ilgili alan yazında, temel iki eğilim bulunmaktadır. Bunlardan ilki, dersi öğrencinin keşfedeceği şekilde tasarlayıp öğrenci merkezli bir yaklaşım sunmaktır. Bu yaklaşıma göre öğrenci süreçte aktif rol üstlenir. Öğretmen bilgiyi aktarmaktan ziyade, öğrencinin keşfetme veya oluşturma sürecinde ona yardım eden pozisyondadır. İkinci yaklaşım ise dersi öğretmenin anlatımı üzerine kurar. Burada öğretmen asıl sorumluluğu alır ve öğrencilere konuyu aktarmayı hedefler. Öğrencinin keşfetmesini temel alan ilk yaklaşımda, öğrenciler belirli bilgi işlemsel problemlerde yetkin olsalar da temel algoritmik kavramlarda ve dolayısıyla bunların farklı bağlamlara aktarımında sıkıntılar yaşamaktadır (Mayer, 2004). Bunun karşısında bilgi aktarımını temel alan bilgi işlemsel öğretim yaklaşımları ise çağımızın öğrenenlerini yetiştirmek için yetersiz kalmaktadır (Papert, 1991). Bu yüzden, alan yazında tartışıldığı üzere (Brown ve Campione, 1994; Grover, Pea ve Cooper, 2015; Mayer, 2004), bu derste temel kavramların oluşturulmasında aktarım yaklaşımı ön plana çıkarılırken, öğrencilerin verilen temel ile ileri seviye zihinsel faaliyetlerde

bulunması ve farklı bağlamlara aktarılabilir bir bilgi oluşturabilmesi için keşfe dayalı yaklaşımlar ön plana çıkarılacaktır. Bu amaç doğrultusunda detayları aşağıda verilecek olan “gözle, uygula, tasarla, üret ve değerlendir” öğrenme döngüsü kullanılmıştır.



Şekil 1. Öğrenme Döngüsü

GÖZLE - UYGULA - TASARLA - ÜRET - DEĞERLENDİR ÖĞRENME DÖNGÜSÜ

Gözle: Bu bölüm iki kısımdan oluşur. Birinci kısımda öğretmen öğrencilerin geçmiş bilgilerini aktive etmek ve onların dikkat ve motivasyonlarını sağlamak ile görevlidir. Bunun için bir önceki derste yapılan etkinlikleri / çalışmalarını kısaca özetleyebileceği gibi günlük yaşamdan ya da bilim insanlarının yaşantılarından ilgi çekici örnekler de kullanılabilir, örneğin otomasyona giriş yapmak için El-Cezeri'nin hayatından ve ürettiği cihazlardan bahsedilebilir. Bu bölümün ikinci kısmında ise öğretmen bir robotik konusunu uygulamalı olarak (göstererek) anlatır. Bu kısımda öğretmen daha aktiftir. Uygulamayı yaparken öğrencilere sorular sorabilir ve öğrencilerin sorularını yanıtlayabilir.

Uygula: Bu bölümde öğretmen öğrenenlerden bir önceki bölümde gösterilen uygulamaların aynısını / bir benzerini ister veya onlarla birlikte yapar. Örneğin, hazırlanmış bir robotun ileri doğru belirli bir hızda ve miktarda hareket ettirilmesini gösteren öğretmen, öğrenenlerden robotu belirli bir hızda ve miktarda ters yönde hareket ettirmesini isteyebilir.

Tasarla: Bu bölümde öğrenciler daha aktif rol üstlenir. Öğretmen rehber pozisyonundadır. Öğretmen, öğrencilere takıldıkları noktalarda destek olacaktır. Öğrencilerin etkinlikten kopup, motivasyonlarının düşmesine izin vermemeye çalışacaktır. Fakat öğretmenin sağladığı destek gereğinden fazla da olmamalıdır. Bu bölümde öğretmen tarafından öğrenenlere bir problem verilir. Öğrenenlerden öncelikle bu problemin çözümünü tasarlamaları istenir. Tasarlama aşamasında, öğrenenler temel itibarıyla bilinenler ile istenenler arasındaki bağı kurarak bir plan üreteceklerdir. Bu amaçla, öğrenenler (Bilgi işlemsel düşünme becerisi bileşenlerini kullanırlar):

- Bilinenleri ve istenilenleri ayrı ayrı belirler,
- İstenilenleri alt bileşenlere ayrılabilirse ayırır,
- Bu problem veya alt problemlerin aynalarına veya benzerlerine daha önceden çözüm ürettiyse bunları tanımlar,
- Bu problemler veya alt problemlerin çözümü için bilgisayar biliminde daha önceden belirlenmiş çözümlerin (sıralama ve arama algoritmaları gibi) olup olmadığını belirler,
- Daha önceki adımlarda ortaya koyduklarını kullanarak bir çözüm planı üretirler.

Bu aşamadaki önemli nokta, öğrenenlerin çözüme doğrudan başlaması yerine önce çözüm hakkında düşünmesi ve bir çözüm planı üretmesidir. Öğrenenler her defasında yukarıda bahsi geçen beş adımı yapmak istemeyebilir veya bu adımları yaparken sıkılabılırlar. Bu durumlarda, öğrenenlerin adımları bire bir uygulaması yönünde onları zorlamak yerine onlardan problemi doğrudan çözmeye başlamadan önce problem hakkında düşünmesi ve planlama yapması istenebilir.

Üret: Bu bölümde öğrenciler aktif rol üstlenir. Öğretmen rehber pozisyonundadır. Öğretmen öğrencilere takıldıkları noktalarda destek olacaktır. Destek, Vygotsky'nin (1978) Yakınsal Gelişim Alanı (Zone of Proximal Development) kavramında belirttiği gibi bireyin yardım ile gerçekleştirebileceği, ancak henüz bağımsız olarak yapamayacağı bir durum olduğunda sağlanmalıdır. Üret aşamasında, öğrenenlerden bir önceki adımda tasarladığı planı kullanarak probleme algoritmik bir çözüm üretmesi istenir. Öğrenenler bilgisayar ve robot başında çalışarak gerekli donanımsal ve yazılımsal çözümleri geliştirirler.

Değerlendir: Buradaki değerlendirme ile anlatılmak istenen doğrudan öğrencinin başarısının notlandırılması değildir. Temel hedef, öğrenenin, öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşünmesini sağlamaktır. Bu sayede, öğrenen problem çözme süreci, dersin konusu ve kendisi ile ilgili gözlemler yaparak yeni öğrenmeler, kendini değerlendirme ve planlama açısından fırsatlar elde edecektir. Öğrenenlerden şu soruları yanıtlamaları istenebilir:

- Verilen problemi tanımlayınız (problemi kendi cümleleri ile ifade etme).
- Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?
- Problemi çözerken ne gibi sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?

- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla ihtilafa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne öğrendiniz?

Öğrenenlerin buradaki soruların tamamına cevap vermesi beklenmemektedir. Bu sorulardan, verilen etkinlikten elde ettikleri deneyimlere bağlı olarak, kendilerine uyanları cevaplayabilirler. Cevaplar, öğrenenlerden yazılı olarak da istenebilir. Fakat öğrenenler, belirli bir süre sonra sürekli aynı sorulara cevap vermekten sıkılabilir/sıkılacaktır. Bu durumda, belirli derslerin sonunda öğrenenlerden dersteki deneyimlerini genel olarak değerlendirmeleri istenebilir. Bunun yanında, değerlendirme sürecini daha etkili hâle getirmek amacıyla çeşitli etkinlikler yapılabilir. Örneğin, öğrenenler bir halka şeklinde dizilir. Öğretmen elinde bulunan topu (veya benzeri bir cismi) öğrenenlerden birine atarak onun yukarıdaki sorulardan bir veya birkaçına yanıt vermesini ya da dersteki deneyimlerini genel olarak değerlendirmesini isteyebilir. Cevap veren öğrenen, elindeki topu bir başka arkadaşına atar ve arkadaşları da bir veya birkaç soruyu yanıtlar. Etkinliklerin ilerleyen zamanlarında, top yerine rastgele açıyla kendi etrafında dönecek şekilde programlanmış olan robot, öğrenci halkasının ortasında bir sonraki öğrencinin seçimini yapabilir. Bu durum, değerlendirme doyum noktasına ulaşana kadar, öğrencileri sıkmadan, devam ettirilebilir.

EŞLİ PROGRAMLAMA

Öğrenenler kendilerine verilen görevlerden uygun olanlarını (yukarıda bahsedilen uygula ve üret adımları) eşli programlama grupları içerisinde yapacaklardır. Eşli programlamada iki öğrenen, bir robot veya bilgisayar karşısında yan yana oturarak tasarım, algoritma, kod yazma veya hata ayıklama için iş birliği çalışır. Eşli programlama eşli araba yarışlarına benzetilebilir. Eşli araba yarışlarında sürücü arabayı kullanırken kılavuz sürücüyeye yön tayini konusunda yardımcı olur. Eşli programlamada bilgisayarı veya robotu kullanan kişiye sürücü denir. Sürücünün görevi robotun istenenleri gerçekleştirilmesi için tasarımı ve kodlamayı yapmaktır. Sürücünün yanındaki kişiye kılavuz denilir. Kılavuz, bilgisayarı veya robotu kullanmaz. Kılavuzun görevi çıkan problemler veya ana problem için çözüm üretmek, bu süreçte ortaya çıkan hataları belirlemek ve sürücünün nasıl çalıştığını değerlendirmektir. Eşli programlamada, araba yarışından farklı olarak, sürücü ve kılavuz düzenli olarak yer değiştirir. Öğrenenlerin her iki görevden de öğreneceği şeyler vardır. Bu yüzden rol değiştirme çok önemlidir. Öğretmen öğrenenlerin periyodik olarak görev değiştirmesini sağlamalıdır.

- İki öğrenen bir bilgisayar veya robot karşısına oturur,
- Bir öğrenen kodları yazarken diğeri kodları değerlendirir ve öneride bulunur,
- Belirli zaman aralıklarıyla öğrenenler rollerini değiştirir.

Eşli programlamada bir diğer önemli nokta, hangi iki öğrencinin eş olarak atanacağıdır. Burada, iyi bilen ve az bilen gibi, farklı bilgi veya beceri gruplarından öğrenenlerin bir araya getirilmesi iyi bilenin az bilene öğretmesi açısından faydalı olarak görülebilir. Fakat pratikte bu fayda gerçekleşmemektedir. Bundan ziyade, iyi bilen bir müddet sonra diğerini kendine ayak bağı olarak görme eğilimi ve görece daha az bilen de iyi bilen ile iletişim kurmakta sıkıntı yaşayıp etkinliklerden kopma eğilimi göstermektedir. Benzer bilgi ve beceri düzeyinde olan gruplar daha verimli çalışmaktadır. Bu yüzden benzer bilgi ve beceri düzeyinde olan öğrenenlerin eş olarak belirlenmesi önemlidir. Bazı durumlarda eş olarak tayin edilen öğrenenler birbirleriyle ciddi anlaşmazlıklar ve çatışmalar yaşayabilirler. Çatışma yaşayan öğrencilerin aynı grupta kalması sağlıklı olmayacaktır. Bu durumu fark eden öğretmen, çatışma yaşayan eşleri farklı öğrenenlerle yeniden eşlemelidir.

GRUPLAR ARASI İLETİŞİM VE ROBOT YARIŞLARI

Etkinlikler boyunca gruplar arası bilgi alışverişine izin verilmelidir. Gruplar arasında paylaşımcı bir ortam oluşturulmalıdır. Fakat bu paylaşım, komple bir çözümün paylaşımı şeklinde olmamalıdır. Öğrenenler; çözüm yolları, stratejiler ve eksik bilgiler gibi konular için paylaşım yapabilirler. Fakat verilen bir problemin bütün çözümü paylaşılmamalıdır. Bu konu hakkında öğretmen, öğrenenleri daha önceden bilgilendirmelidir ve onların ne tür bir paylaşım içerisinde olduğunu takip ederek gereğinden fazla olan paylaşımları engellemelidir.

Robot yarışları, öğrenenleri güdüleyen önemli bir öğretme/öğrenme yöntemidir. Bu eğitim programı içerisinde zaman zaman gruplar arası yarışlar düzenlenmiştir. Fakat bu yarışlar, sınıfın paylaşımcı ortamını zedeleyecek içerikte uygulanmamalıdır.

ROBOTİK KİTLER VE PROGRAMLAMA ORTAMI

Lego Mindstorms Education EV3

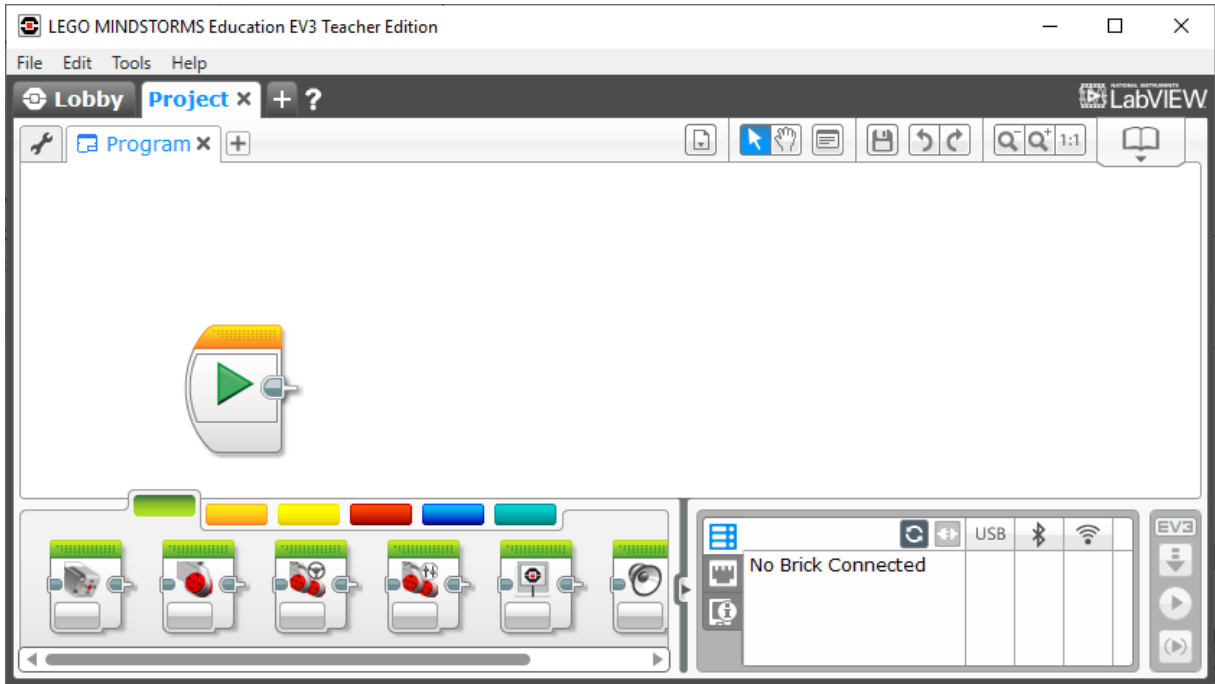
Bu eğitimde LEGO firmasının 2013 yılında piyasaya sürdüğü Mindstorms Education EV3 sürümü kullanılacaktır. 541 parçadan oluşan bu setin içinde EV3 programlanabilir tuğla, renk sensörü, ultrasonik sensör, dokunma sensörü ve jiroskop sensör dâhil birçok teknik parça mevcuttur. Temel setle ve temel set + ekstra parça seti ile inşa edebilen farklı robot tasarımları mümkündür. Aşağıdaki resimde EV3 robot seti görülmektedir.



Resim 1. LEGO® MINDSTORMS® Education EV3 robot seti

EV3 Yazılımı

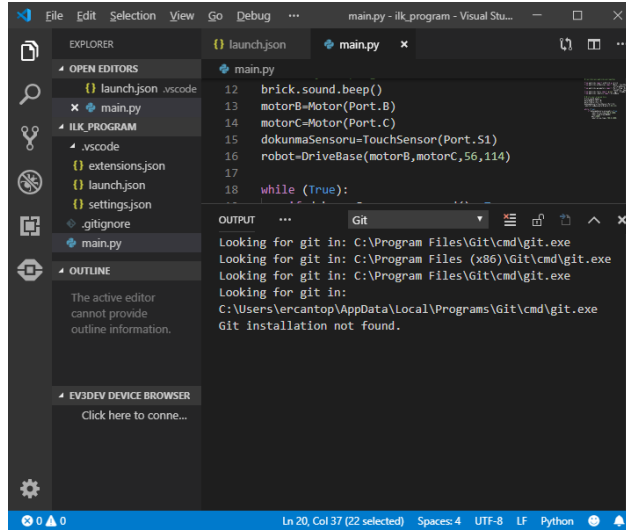
EV3 yazılımı robotları programlamak için kullanılan ücretsiz bir blok tabanlı programlama ortamıdır. Öğrenciler blok hâlinde var olan veya kendilerinin oluşturdukları komutları sürükleyip bırak yöntemiyle taşıyarak programları oluştururlar. Komutlar bloklar hâlinde bulunduğu için, öğrencilerin söz dizimini öğrenmek için fazladan zaman ayırmaları gerekmez ve program yazarken hata yapma ihtimalleri ortadan kaldırılır. Bu sayede öğrenciler zihinsel kaynaklarını problem çözmeye yönlendirebilirler. EV3 yazılımının programlama arayüzü aşağıdaki resimde görüldüğü gibidir.



Resim 2. EV3 Yazılımı

MicroPython

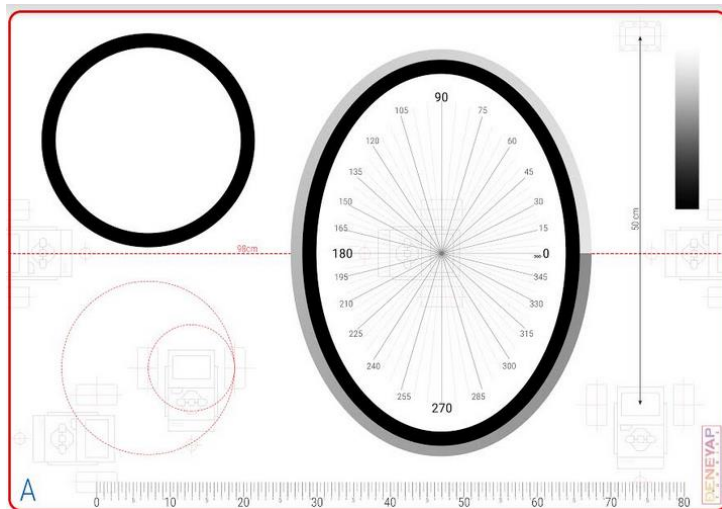
“MicroPython” sayesinde “Python” programlama dili ile Lego Mindstorms EV3 robotu programlanabilir. Burada metin tabanlı programlama yaklaşımı kullanılmaktadır. MicroPython’da öğrenciler yazacakları her bir komutu en ince detayına kadar tam olarak bilmek zorundadırlar. Bu yüzden robotik programlamaya giriş yapmak EV3 yazılımına göre daha zor olmaktadır. Fakat metin tabanlı programlama yaklaşımı profesyonel programcılar tarafından yıllardır kullanılmaktadır. Metin tabanlı programlama yaklaşımına alıştıktan sonra karmaşık / ileri seviye programların oluşturulması bu ortamlarda daha kolaydır. Aşağıdaki resimde görülen “Visual Studio Code” programlama editörü kullanılarak MicroPython’da programlar yazılabilir.



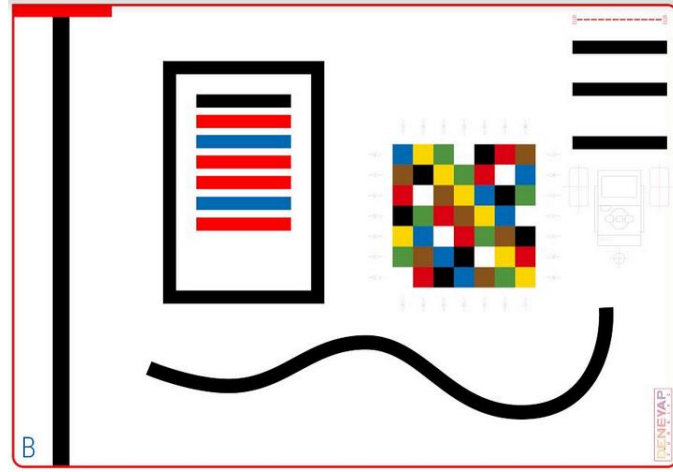
Resim 3. Visual Studio Code Ortamı

Eğitimde Kullanılacak Mat (Çalışma Alanı)

Öğrenciler etkinlik alanına geldiklerinde aşağıda resimleri verilen 70 cm x 100 cm boyutundaki çift taraflı matı (çalışma alanını) masalarının üzerinde görecektir. Etkinlikler boyunca bu mat kullanılacaktır.



Resim 4. Mat Ön Yüz



Resim 5. Mat Arka Yüz

Kaynakça

- Amerikan Ulusal Araştırma Konseyi (National Research Council). (2012). *Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century*. Washington, DC: The National Academies Press. <https://doi.org/10.17226/13398>.
- Brown, A. L., & Campione, J. C. (1994). Guided discovery in a community of learners. In K. McGilly (Ed.), *Classroom lessons: Integrating cognitive theory and classroom practice* (pp. 229–272). Cambridge: MIT Press.
- Çetin, İ. (2013). Visualization: a tool for enhancing students' concept images of basic object-oriented concepts. *Computer Science Education*, 23(1), 1–23.
- Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199–237.
- Jonassen, D. (2011). Designing for problem solving. In R. Reiser & J. Dempsey (Eds.), *Trends and Issues in Instructional Design and Technology*, (pp. 64–74). Boston, MA: Pearson Education.
- Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American psychologist*, 59(1), 14–19.
- Merrill, M. D. (2007). A task-centered instructional strategy. *Journal of Research on Technology in Education*, 40(1), 5–22.
- Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1–11). NJ: Ablex.
- Papert, S. (1993). *Mindstorms: children, computers, and powerful ideas* (2nd ed.). New York: Basic Books.
- Vygotsky, L. (1978). Interaction between learning and development. *Readings on the Development of Children*, 23(3), 34–41.

1. Hafta: Robot ve Algoritma

Ön Bilgi:

- Temel bilgisayar ve donanım bilgisi

Haftanın Kazanımları:

- Öğrenciler robot kavramını temel düzeyde açıklar.
- Öğrenciler algoritma kavramını açıklar.
- Öğrenciler basit bir algoritma örneği oluşturabilir.
- Öğrenciler robot setini tanır.
- Öğrenciler robot setiyle yönergeye uygun robot oluşturur.

Haftanın Amacı:

Bu haftanın amacı, “robot” kavramının tüm öğrenciler tarafından doğru bir şekilde anlaşılmasını sağlamaktır. Haftanın bir diğer amacı, örnek robotlar göstererek öğrencilerin ilgisini çekmek ve etkinliklere yönelik motivasyonlarını artırmaktır. Ayrıca, ders süresince kullanılacak robot setini tanıtmak ve bu seti kullanarak mekanik robot tasarımları yapmaktır. Öğrencilerin algoritmanın ne olduğunu tanımlaması ve basit bir algoritma oluşturabilmesi de hedeflenmektedir.

Kullanılacak Malzemeler:

Robot seti, bilgisayar, kâğıt, kalem

Haftanın İşlenişi:

Gözle: Robot kavramı üzerine tartışma ve robot tasarlama

Uygula: Robot yapma ve algoritma oyunları

Tasarla: “Hayalimdeki Robot” etkinliği

Üret: Örnek video gösterimi ve “İlk Robotumu Yapıyorum” etkinliği

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği

1. GÖZLE VE UYGULA

1.1. Gözle: Robot Kavramı Üzerine Tartışma

Rehber öğretmen, öğrencilerin "robot" kavramı üzerinde tartışmalarını sağlar.

- Robot nedir?
- Robotların özellikleri nelerdir?
- Robotlar nerelerde kullanılır?
- Uzaktan kumandalı bir oyuncak ile robot arasında ne gibi farklılıklar vardır?
- Robotlar hangi parçalardan oluşur?
- Mekanik nedir?
- Program/algoritma nedir?

gibi sorular yöneltmek tartışmayı yönetir. Cevapları öğrencilerin bulması sağlanır. Rehber öğretmen; robot, algoritma ve program kavramları için aşağıdaki tanımları dikkate alarak öğrencileri yönlendirebilir veya gerektiği yerde (öğrenciler kavramı yanlış tanımladıklarında veya doğru olandan çok farklı bir tanım yaptıklarında) bu tanımları doğrudan kendisi yapabilir.

Robot, Türk Dil Kurumunun hazırladığı *Türkçe Sözlük*'te “belirli bir işi yerine getirmek için manyetizma ile kendisine çeşitli işler yaptırılabilen otomatik araç” şeklinde tanımlanır. *Oxford İngilizce Sözlük*'te ise “özellikle bir bilgisayar tarafından programlanan, karmaşık bir dizi işlemi otomatik olarak gerçekleştirebilen makine” şeklinde tanımlanır. Her iki tanım da vurgulanan nokta, robotların bir görevi otomatik olarak yerine getirmesidir. Eğer bir makinenin hareketleri bir insan tarafından kontrol ediliyorsa bu makineye robot denilemez. Ama makine çevresini algılayabiliyor ve buna göre hareket ediyorsa, örneğin bir engeli algılayıp yön değiştirebiliyorsa, bu makine robot olarak adlandırılır (Kelly, 2010).

Robotlar çevrelerini sensörleri aracılığıyla algılar. Robotlar kullanılan sensöre bağlı olarak; nesnenin uzaklığı, nesnenin rengi, ışık miktarı, ses şiddeti, nem oranı gibi birçok çevresel veriyi algılayıp işlemcileri ile yorumlayabilir ve programları dâhilinde bu verilere tepkide bulunabilirler. Bu durum göz önüne alındığında robotun tanımı, “çevresini algılayabilen ve algıladığını yorumlayarak bağımsız tepkiler verebilen makine” şeklinde yapılabilir. Bir makinenin robot olup olmadığını değerlendirmek için aşağıdaki özellikleri taşıyıp taşımadığına bakılabilir:

- Sensör: Çevresini sensörü/sensörleri aracılığıyla algılayabiliyor mu?
- Program: Bir programı var mı? Programı ile verileri işleyip karar verebiliyor mu?
- Eylem: Programı sayesinde farklı eylemler gerçekleştirebiliyor mu?

Algoritma: Sonlu bir süre içerisinde istenilen bir sonucu elde etmek için açık ve net bir şekilde tanımlanmış ve sıralanmış adımlar bütünüdür. Aslında her öğrenci matematik derslerinde algoritmaları kullanmıştır. Örneğin alt alta eldeli toplama bir algoritmadır. Bu işlemin sonucunu bulmak için uygulanacak adımlar kişiden kişiye değişmeyecek netliktedir. Bu adımlar verildiğinde alt alta eldeli toplamayı uygulayan herkes aynı sonuca ulaşır. Peki, alt alta eldeli toplamamanın adımları nelerdir? Burada öğrencilerden temel olarak şunları söylemeleri beklenir:

- (i) sayıların alt alta yazılması
- (ii) kolonlardaki sayıların toplanması ve sonuç olarak yazılması
- (iii) elde değeri olduğunda bunun bir soldaki kolonun toplama değerine eklenmesi
- (iv) ilk üç kuralın toplama işlemi tamamlanana kadar devam ettirilmesi.

Program: Temel olarak bir algoritmanın bilgisayar veya robot için uygulanmasıdır. Programlar bilgisayara veya robota yapmaları gereken şeyleri (algoritmanın adımlarını) bir programlama dili vasıtası ile aktarırlar. Öğrenciler de bu derste robotları kodlamak için bir programlama dili kullanacaktır. Programlamanın en önemli noktasıysa problemin çözümü için algoritma geliştirmektir. İzlenecek yol doğru belirlenebilirse yani algoritma doğru hazırlanabilirse bilgisayar ve robotlar programlama dilleri aracılığıyla problemi çözebilir. Bu bilgilendirmeler yapıldıktan sonra algoritma uygulamaları bütün öğrencilerle beraber yapılır.

1.2. Uygula: Algoritma Öğreniyorum

Sınıftaki öğrencilere üzerinde 1 ile 24 arasındaki sayıların yazılı olduğu kartlar verilir. Öğrenci sayısı 24'ten azsa 1, 2, 3, 4, 6, 8, 12, 24 sayılarını mutlaka içeren, istenilen sayıda kart dağıtılır. Öğrenci sayısı 24 ise her öğrenciye bir kart verilir. Öğrenci sayısı 24'ten fazlaysa birden fazla öğrenciye aynı sayı gelecek şekilde kartlar ayarlanır. Dağıtılan kartlar arasında yine 1, 2, 3, 4, 6, 8, 12, 24 sayılarının hepsi bulunmalıdır. Öğrenciler kartlarındaki sayıyı saklayarak grup hâlinde bekler. Öğrencilerden biri "robot" olarak görevlendirilir ve sadece bu "robot-öğrenciye" diğerleri duymayacak şekilde aşağıdaki algoritma verilir:

- 1) Her bir öğrenciye git,
 - a) Sayısını sor,
 - b) Eğer öğrencinin kartındaki sayı 24'ü tam bölen bir sayıysa bu öğrenciyi gruptan ayır ve tahtanın önüne getir.

Robot-öğrenci diğer öğrencilere sadece kartlarındaki sayıyı sormalıdır ve ipucu oluşturacak cümleler kurmamalıdır. Robot-öğrencinin işlemini tamamlamasının ardından rehber öğretmen, diğer öğrencilerden, robot-öğrencinin tahtaya çıkardığı öğrencilerin sayılarına bakarak hangi işlemin yapılmış olabileceğini söylemelerini ister. Öğrenciler yapılan işlemi bildikten sonra onlardan bu işlemin algoritmasını yazmaları istenir.

1.3. Uygula: Çarpım Eşleri Algoritması

Öğrencilerden, çarpımları 24 olan çarpım eşlerinin (1-24, 2-12, 3-8, 4-6) bulunması için bir algoritma yazmaları istenir. Öğrenciler buldukları algoritmaları sınıfta tartışabilirler.

Son olarak algoritmanın bir görevi gerçekleştirilmesi için net bir şekilde tanımlanmış ve sıralanmış adımlardan oluşması gerektiği vurgulanır. Eğer görevleri açıkça tanımlanmadıysa veya adımların sıralaması uygun değilse robotun veya bilgisayarın istenilen görevi yerine getiremeyeceği belirtilir.

2. TASARLA

2.1. “Hayalimdeki Robot” Etkinliđi

Öğrencilerden belirli bir amaç için tasarlanmış bir robot hayal etmeleri ve grup olarak hayal edilen robotun resmini çizmeleri istenir. Daha sonra gruplar çizimleri üzerinden robotlarını arkadaşlarına tanıtır ve robotun ne amaçla kullanılacağını anlatırlar.

Gerekli malzemeler:

1. Her bir grup için beyaz A3 kâğıt
2. Boya kalemleri
3. Makas
4. Yapıştırıcı

Bu aşamada öğrenciler gruplara ayrılır. Her gruptan günlük hayatta karşılaşılan bazı problemleri düşünmeleri, bu problemleri çözebilecek bir robot hayal etmeleri istenir. Daha sonra öğrencilerden hayal ettikleri robotları kâğıtlara çizmeleri ve kesmeleri istenir. Öğrencilere 20 dakika süre verilir. Süre bittikten sonra her grup tek tek tahtaya çıkararak kendi robotunu tanıtır. Hayal edilen robotlar panoya asılır ve eğitim süresi boyunca panoda bırakılır.

3. ÜRET

3.1. “İlk Robotumu Yapıyorum” Etkinliđi

Öğrencilere eğitim süresince neler başarabileceklerini göstermek için robot setleri kullanılarak yapılmış robotların videoları gösterilir. Öğrencilere robot setleri ile çalışırken nelere dikkat etmeleri gerektiđi ile ilgili bilgilendirme yapılır.

Örnek videolar:

- EV3 Stair Climber <https://www.youtube.com/watch?v=BZSkFI2wPzk>
- EV3 Robot Arm <https://www.youtube.com/watch?v=9cTo20lmyKY>
- EV3 Gyro Boy <https://www.youtube.com/watch?v=5pinA9k1E9g>
- EV3 Spinning Top Factory <https://www.youtube.com/watch?v=0yvFlq0wGJA>

Rehber öğretmen öğrencilere LEGO MINDSTORMS EV3 setini tanıtır. Robotu yaparken dikkat edilmesi gereken noktaları belirtir. Parçaların sağlam olduğunu fakat düşme, çarpma gibi durumlar sonucu kırılıp yıpranabilecekleri ifade eder. Parçaların kaybolmaması için robot seti ile gelen veya var olan tasnif kutusunun kullanılmasını önerir.

Her gruba bir robot seti dağıtılır. Beş dakika kadar öğrencilerden robot setlerini incelemeleri, robot setlerindeki parçaları ve ne amaçla kullanılacaklarını keşfetmeleri beklenir. Ardından öğrenciler bu konu hakkındaki düşüncelerini paylaşır. Gerekli noktalarda rehber öğretmen temel parçaların nasıl kullanılacağıyla ilgili daha detaylı bilgiler verir.

Öğrencilerden “temel tasarım (driving base)” yönergesine uyararak ilk robotlarını oluşturmaları istenir. Öğrenciler grup çalışması sırasında görev paylaşımı yapmaları konusunda cesaretlendirilir. Daha sonra öğrencilere robot setleriyle temel tasarım robotunu yapmaları için yeteri kadar süre verilir.

Not

Temel tasarım yönergesine aşağıdaki yollarla ulaşılabilir:

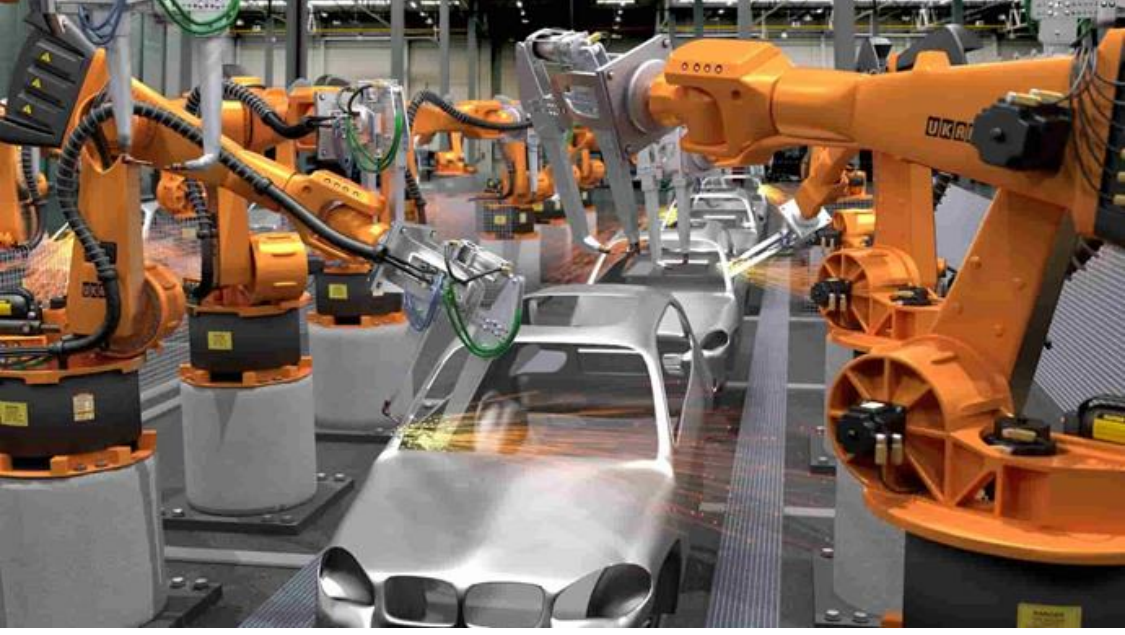
- i) <https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-rem-driving-base-79bebf16bd491186ea9c9069842155e.pdf>
- ii) EV3 yazılımı > Lobby > Building Instructions > Building Ideas > Driving Base
- iii) Robot setiyle birlikte gelen kitapçığa bakılabilir.

Rehber öğretmen, robotlar oluşturulduktan sonra, daha önce hazırladığı bir programı robot üzerinde çalıştırarak gösterir. Buradaki amaç, öğrencilere kod yazmayı öğretmek, yazılan kodu robota yüklemek veya yüklenen kodun robot üzerinde nasıl çalıştırılacağını göstermek değildir. Amaç, robota basit bir görev yaptırarak öğrencilerin robota karşı olan ilgilerini artırmaktır.

4. DEĞERLENDİR

Öğrenciler ile halka oluşturulur. Rehber öğretmen; robot, robot tasarımı, algoritma ve program konularından birini seçer ve rastgele seçtiği bir öğrenciden verilen kavramın onun için ne ifade ettiğini anlatmasını ister. Bütün konuların anlatılmasının ardından, öğrencilerden, robot tasarlama ve algoritma geliştirme aşamalarında önemli veya ilginç buldukları noktaları arkadaşlarıyla paylaşmaları istenir.

Daha sonra aşağıdaki resim gösterilerek ilgili sorular hakkında öğrencilerin görüşleri alınır.



Resim 6. Endüstriyel Robotlar

- Bu robotun görevleri nelerdir?
- Bu robotu kullanarak hangi sorunları çözebiliriz?
- Sizce mutfak robotu, derste işlediğimiz teknik anlamıyla, gerçek bir robot mudur? Sebeplerini açıklayınız.

5. İLAVE ETKİNLİK

5.1. Algoritma oyunu (Sırt sırta oturup ayakkabı bağcığı bağlama)

Bu oyununun oynatılmasındaki amaç, öğrencilerin bir görevi gerçekleştirmek için gerekli adımları net bir şekilde tanımlamaları ve sıralamaları gerektiğini görmelerini sağlamaktır. Öğrenciler görevleri açıkça tanımlamazlarsa veya adımların sıralamasını uygun yapmazlarsa istenilen görevin yerine getirilemediğini oyun sonunda görürler.

Oyun için sınıftan ayakkabısı bağcıklı olan iki öğrenci seçilir. Oyun için seçilen öğrenciler diğer öğrencilerin görebileceği bir yere sırt-sırta oturtulur. Öğrencilerin ayakkabı bağları çözdürülür. Bir öğrenciye ayakkabıyı bağlarken arkadaşına da detaylı olarak anlatması görevi verilir. Diğer öğrenciye de arkadaşının anlattığı şekilde kendi ayakkabısını bağlama görevi verilir. Dinleyen öğrenciye sadece arkadaşının anlattığı şekilde adımları yapması, kendisinin ilave işlem yapmaması gerektiği söylenir. Bu oyunda, anlatan öğrencinin kendi ayakkabı bağcığını bağlarken yaptığı adımları arkadaşına aktarırken yeterince açık ve bilgilendirici olmadığı diğer öğrenciler tarafından görülür. Oyunu oynayan öğrenciler de görevlerin net bir şekilde tanımlanması gerektiğini deneyimlemiş olurlar. Oyun ders süresi bitinceye kadar tekrar edilir.

2. Hafta: Robotlar ile Hareket

Ön Bilgi:

- Öğrenciler robot kavramını temel düzeyde bilir.
- Öğrenciler robot setiyle temel robotik tasarımı yapmıştır.

Haftanın Kazanımları:

- Öğrenciler robot setini programlamak için EV3 yazılımının grafik arayüzünü kullanır.
- Öğrenciler açı, tur sayısı, çap (yarıçap), çevre ve uzunluk kavramlarını robotun hareketini sağlarken kullanır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturur.

Haftanın Amacı:

Bu haftanın amacı, robot seti ile ve robot setinin programlanacağı EV3 yazılımının grafiksel arayüzünü tanıtmaktır. Ayrıca öğrencilere robotlarının belirli bir mesafeyi alabilmesi için gerekli programlama adımlarını öğretmek ve öğrencilerin ölçme, uzunluk, denklem, çap ve çevre konularını verilen problemin çözümünde kullanmalarını sağlamaktır.

Kullanılacak Malzemeler:

Robot seti, bilgisayar, mat (çalışma alanı)

Haftanın İşlenişi:

Gözle: Robotun bileşenlerini tanıma ve robot setinin programlanacağı yazılımın grafiksel arayüzünü kullanma

Uygula: Robot yapma ve robot programlama

Tasarla: İstenilen robotu tasarlama

Üret: Verilen görevleri programlama

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği

1. GÖZLE VE UYGULA

1.1. Gözle: EV3 Yazılımı

İlk hafta hazırlanan robot setleri öğrencilere dağıtılır. Hazır bir robot üzerinde robot setinin parçaları gösterilerek öğrencilere parçalar hakkında temel bilgiler verilir. Böylece ilk hafta konuları tekrar edilmiş olur. Daha sonra EV3 robot setlerinin programlanması için gerekli yazılımın bilgisayarlarda kurulu olup olmadığı kontrol edilir. LEGO MINDSTORMS EV3 robot setinin akıllı tuğla (intelligent brick) üzerinden de programlanabildiğinden, ancak eğitim süresince robotların programlanması için bilgisayarların kullanılacağından bahsedilir.

Not

Lego Mindstorms Education EV3 yazılımına

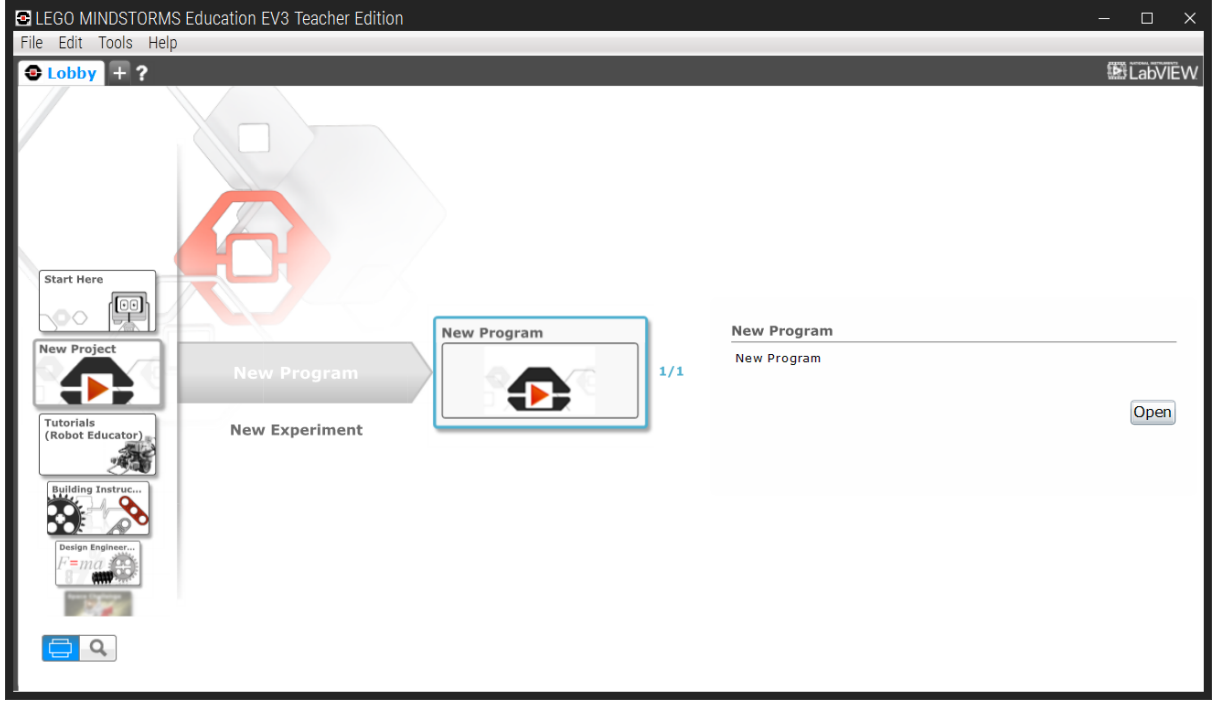
<https://education.lego.com/en-us/downloads/retiredproducts/mindstorms-ev3-lab/software>

adresinden uygun işletim sistemi seçilerek veya Windows işletim sistemi için aşağıdaki adresten direkt ulaşılabilir.

https://le-www-live-s.legocdn.com/downloads/LME-EV3/LME-EV3_Full-setup_1.4.5_en-US_WIN32.exe

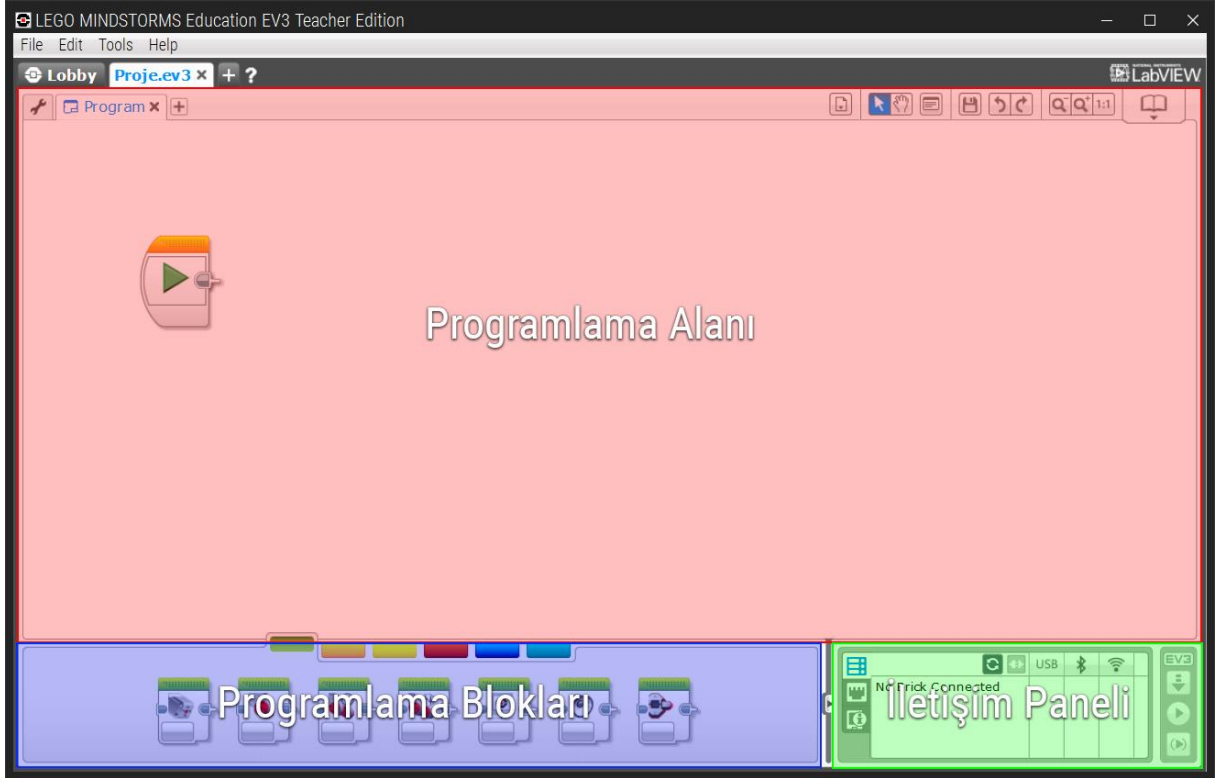
Robot setinin programlanacağı yazılımın grafiksel arayüzü (EV3 yazılımı) açılır ve arayüzde neler olduğu gösterilir. Grafik arayüzü ile akıllı tuğla arasındaki bağlantının USB veya Bluetooth ile gerçekleştirilebileceği vurgulanır. Önce kablo ile sonra da Bluetooth ile bağlantıların nasıl yapılacağı anlatılır. Daha sonra EV3 yazılımının grafiksel arayüzünün bölümleri tanıtılır.

Aşağıdaki resimde görüldüğü gibi EV3 yazılımı Lobby ekranı ile açılır. Bu ekranda sol kısımdaki etkinlik sekmeleri kullanılarak yeni bir program penceresi açılır (New Project > New Program > Open).



Resim 7. EV3 Yazılımı Lobby Ekranı

EV3 yazılımının arayüzü temel olarak üç kısımdan oluşur. Robotun programlanabilmesi için program bloklarının programlama alanına sürüklenip bırakılması gerekir. Programlama alanında varsayılan olarak aşağıdaki resimde görüldüğü gibi bir başla bloğu (yeşil üçgen) bulunur. Programlama alanında birbirine yaklaştırılan bloklar birbirine yapışır. Böylece sürükle-bırak yaklaşımı ile istenilen programın oluşturulması sağlanır.



Resim 8. EV3 Yazılımı Programlama Arayüzü

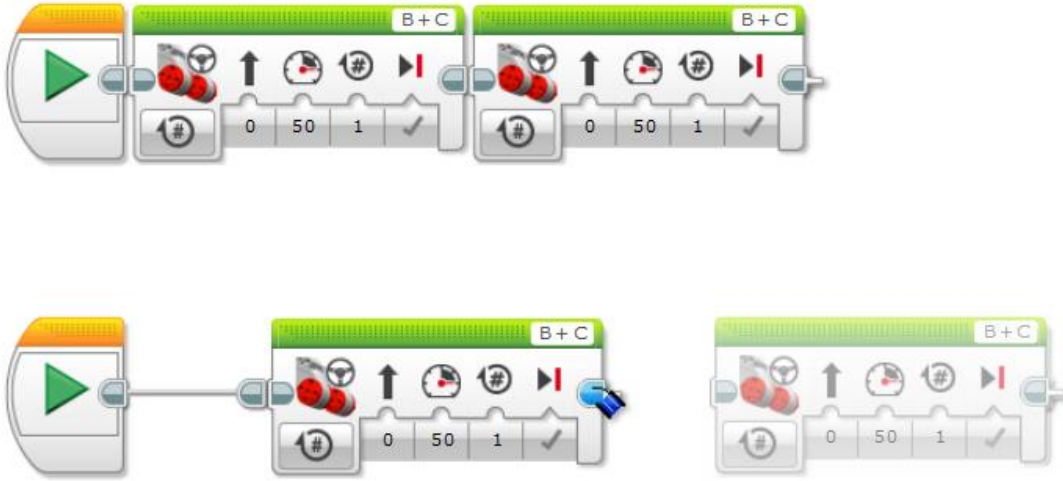
Yukarıdaki resimde görüldüğü gibi programlama blokları bölümü farklı renklerdeki altı sekmeden oluşur:

- Yeşil Sekme (Action/Eylem Sekmesi): Robotun hareket etmesini sağlayan motor kontrol blokları, görüntü, ses ve tuğla ışığı blokları bu sekmede yer alır.
- Turuncu Sekme (Flow Control/Akış Kontrol Sekmesi): Yeni bir başla program bloğunun yanı sıra bekle, döngü, anahtar ve döngü kesme blokları bu sekmede bulunur.
- Sarı Sekme (Sensor/Sensör Sekmesi): Robotun sensörlerinden ışık, uzaklık, motorların dönme açısı gibi verilerin alınmasını sağlayan program blokları bu sekmede yer alır.
- Kırmızı Sekme (Data Operations/Veri İşlemleri): Veriler üzerinde matematiksel ve mantıksal işlemlerin gerçekleştirilebileceği blokların ve değişken, sabit ve rastgele sayı üreten blokların bulunduğu sekmedir.
- Mavi Sekme (Advanced/İleri Düzey Sekmesi): İleri düzey programlama bloklarının, örneğin dosya erişimi, veri kaydı, Bluetooth bağlantısı gibi blokların bulunduğu sekmedir.
- Turkuaz Sekme (My Blocks/Bloklarım Sekmesi): Program birçok bloktan oluşuyorsa belirli görevleri yerine getiren program blokları bir blok olarak tanımlanabilir. Tanımlanan bu blok gruplarına benim bloklarım sekmesinden ulaşılabilir.

İletişim paneli üzerinden USB, Bluetooth veya kablosuz bağlantı ile bağlı olan akıllı tuğlanın durumu ve sensörlerden alınan anlık veriler takip edilebilir. Ayrıca programlama alanında hazırlanmış programın tüm bloklarının ya da seçili blokların akıllı tuğlaya yüklenip çalıştırılmasını sağlayan düğmeler de bu bölümde yer alır.

1.2. Gözle: Programlama Alanının Kullanılması

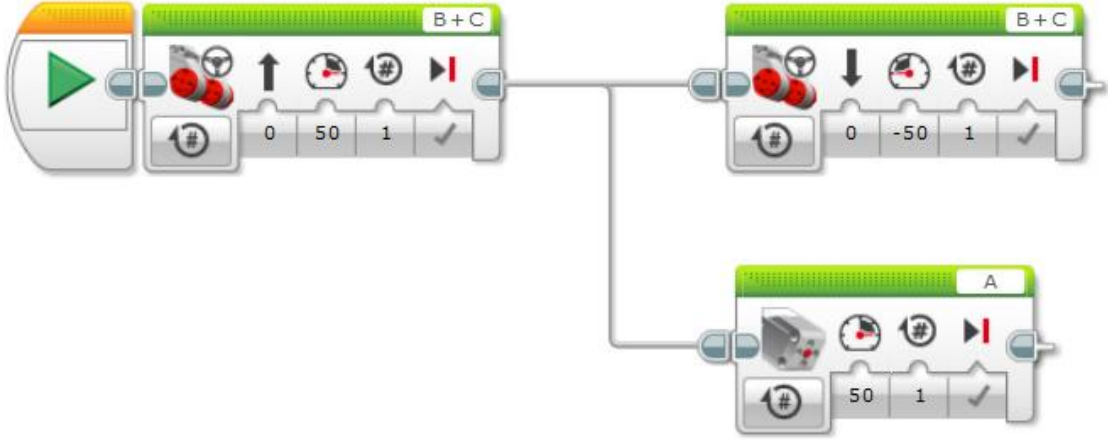
Robot, istenilen program bloğunun programlama alanına sürüklenip bırakılması ve blokların parametrelerinin değiştirilmesi ile programlanır. Program, blokların programlama alanında yer alan başla bloğuna tren vagonları gibi birbiri ardına eklenmesiyle oluşturulabilir. Program çalıştırıldığında, robot, başla bloğundan başlayarak program bloklarında belirlenen görevleri sırasıyla gerçekleştirecektir. Aşağıdaki resimde blokların yerleştirilmesi görülmektedir.



Resim 9. Blokların Yerleştirilmesi

Başla bloğu ile herhangi bir bağlantısı bulunmayan bloklar daha soluk gösterilir ve programın çalıştırılması sırasında programa dâhil edilmez. Bloklar arasındaki bağlantı, blokların yan yana eklenmesi ile sağlanabileceği gibi bir bloğun bittiği noktadan diğer bloğa bir kablo bağlantısı yapılarak da gerçekleştirilebilir. Bunun için fare bloğun bittiği noktanın üzerine getirilmeli ve imlecin değişmesinin ardından sürüklenip istenilen noktada bırakılmalıdır.

Robotun birden fazla işlemi aynı anda gerçekleştirmesi isteniyorsa, programlama alanında birden fazla başla bloğu kullanılarak, iki farklı programın eş zamanlı çalışması sağlanabilir. Ayrıca kablo yöntemi kullanılarak programların istenilen noktada paralel işlemler başlatılabilir (Paralel işlemler ilerleyen haftalarda daha detaylı işlenecektir). Paralel işlemler aşağıdaki resimde görülmektedir.



Resim 10. Paralel İşlemler

1.3. Gözle: Robotun Hareket Ettirilmesi

EV3 robot setinde robotu hareket ettiren iki çeşit motor bulunur. Bunlar aşağıdaki resimde görüldüğü gibi büyük (large) ve orta (medium) motor olarak adlandırılır. Set içerisinde iki büyük motor ve bir orta motor mevcuttur. Büyük motor robotu hareket ettirmek için kullanılır ve orta motora göre daha güçlüdür. Fakat orta motor büyük motordan daha hızlı çalışır. Her iki motorun içerisinde devir sensörü bulunur. Devir sensörü kullanılarak motorun devir sayısı bulunabilir. Motorun devir sayısı açılı ile hesaplanabileceği gibi bir tam devir temel alınarak kaç devir döndüğü de bulunabilir.



Resim 11. Büyük ve Orta Motor

Not

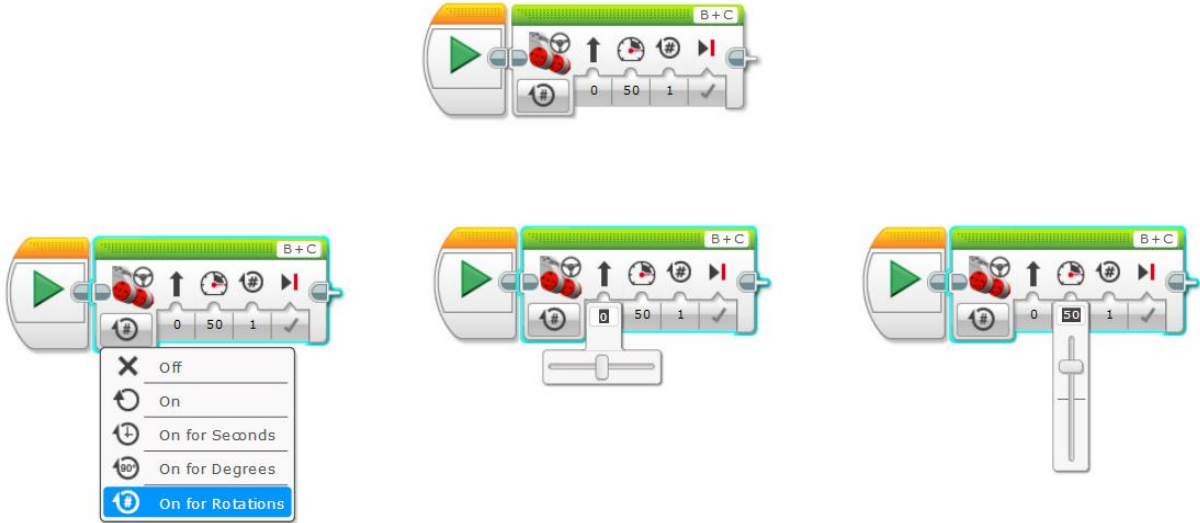
Öğrencilerin önceki hafta oluşturdukları temel tasarım (driving base) robotunun yeterli düzeyde şarj olmuş şekilde sınıfta bulunması gerekir. Ayrıca robotun sol büyük motorunun B portuna, sağ büyük motorunun C portuna bağlandığından emin olunmalıdır.

1.4. Hareket Blokları

Robotu hareket ettirecek program blokları yeşil sekmede (Action) yer alır. Orta motor, büyük motor, direksiyon hareketi (move steering) ve palet hareketi (tank move) olmak üzere dört farklı hareket bloğu bulunur.

1.4.1. Gözle: Direksiyon Hareketi (Move Steering)

Direksiyon hareketi bloğu robotun iki motorunu aynı anda kontrol eder. Her iki motorun pozitif veya negatif yönde dönmesini, dolayısıyla robotun ileri veya geri gitmesini sağlar. Robotun dönmesi ise motorlardan birinin daha hızlı dönmesi ile sağlanır. Bloğun sağ üst köşesinde (örneğin aşağıdaki resimlerde B+C olduğu gibi) hangi portlara bağlı olan motorların kontrol edileceği görülebilir. Eğer motorlar farklı portlara bağlanmışsa bu kısma tıklanarak portlar değiştirilebilir.



Resim 12. Direksiyon Hareketi Bloğu ve Ayarlanması

Öncelikle motorların çalışma modu seçilir (sol alttaki resim): motorların durdurulması (off), sürekli (on), belirli bir süre (on for second), belirli bir açı değerinde (on for degrees)

ve belirli bir tur sayısı (on for rotation). Seçilen moda bağlı olarak blok girişleri değişecektir.

Robotun sağa veya sola dönmesi direksiyon değerinin -100 ile 100 arasında değiştirilmesi ile sağlanır (altta ortadaki resim). Robotun hızı güç değerinin -100 ile 100 arasında değiştirilmesi ile azaltılabilir ya da artırılabilir (sağ alttaki resim). Negatif değerler robotun geri gitmesini sağlayacaktır. Seçilen moda bağlı olarak (süre, açı, tur) motorların ne kadar döneceği ayarlanabilir. Son olarak motorlar dönme hareketini gerçekleştirdikten sonra fren yapılarak durdurulabilir veya boşa alınabilir.

Not

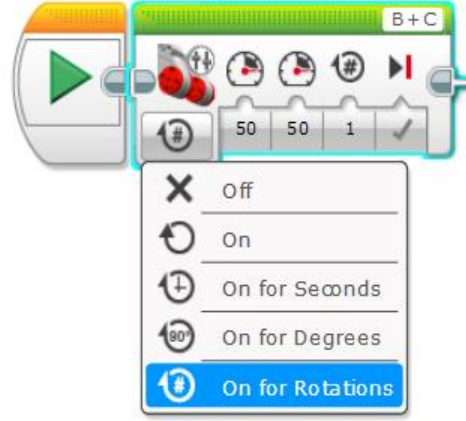
Özellikle derece modunda çalışırken, motorların 180 derece dönecek şekilde programlanmasının, robotun 180 derece dönmesi olarak algılanmaması gerektiği vurgulanmalıdır. Öğrencilerle bu durum tartışılarak oluşabilecek yanlış algı önlenmelidir.

1.4.2. Uygula: Tekerin Yarıçapını Hesaplama

Direksiyon hareketi bloğu anlatıldıktan sonra robotun bir miktar düz gidip kendi etrafında dönmeden geri geldiği bir program çalıştırılıp gösterilir. Öğrencilerden benzer bir uygulamayı tur, saniye ve açı modlarını kullanarak yapmaları istenir. Daha sonra masalarındaki matı kullanarak tekerin bir tur döndürüldüğünde aldığı mesafeyi cm cinsinden bulmaları beklenir. Etkinlik sonunda öğrencilerden tekerin yarıçapını hesaplamaları istenir.

1.4.3. Gözle: Palet Hareketi (Move Tank)

Palet hareketi bloğu, direksiyon hareketi bloğuna hayli benzer. Palet hareketi bloğu iki motorun birbirinden bağımsız olarak farklı güç değerlerinde çalıştırılabilmesine imkân sağlar. Böylece robot olduğu yerde (merkezi etrafında) veya bir tekerinin etrafında dönebilir.



Resim 13. Palet Hareketi Bloğu

1.4.4. Uygula: Palet Hareketi

Palet hareketi bloğu anlatıldıktan sonra öğrencilerden bu bloğu kullanarak robotun dönmesini sağlayan bir program oluşturmaları istenir. Motorların güç değeri arasındaki fark arttıkça robotun dönüşünde meydana gelen değişikliği gözlemlenmeleri beklenir. Öğrenciler robotun olduğu yerde veya bir tekerinin etrafında dönebilmesi için motor güçlerinin nasıl ayarlanması gerektiğini, mat üzerinde yer alan kesikli kırmızı çizgilerle belirtilen daireleri kullanarak keşfeder. Öğrencilerden robotlarını belirli bir mesafe (örneğin 3 tur) düz gittikten sonra kendi etrafında dönüp tekrar başlangıç noktasına gelecek şekilde programlamaları istenir.

1.4.5. Gözle: Büyük Motor (Large) ve Orta (Medium) Motor Blokları

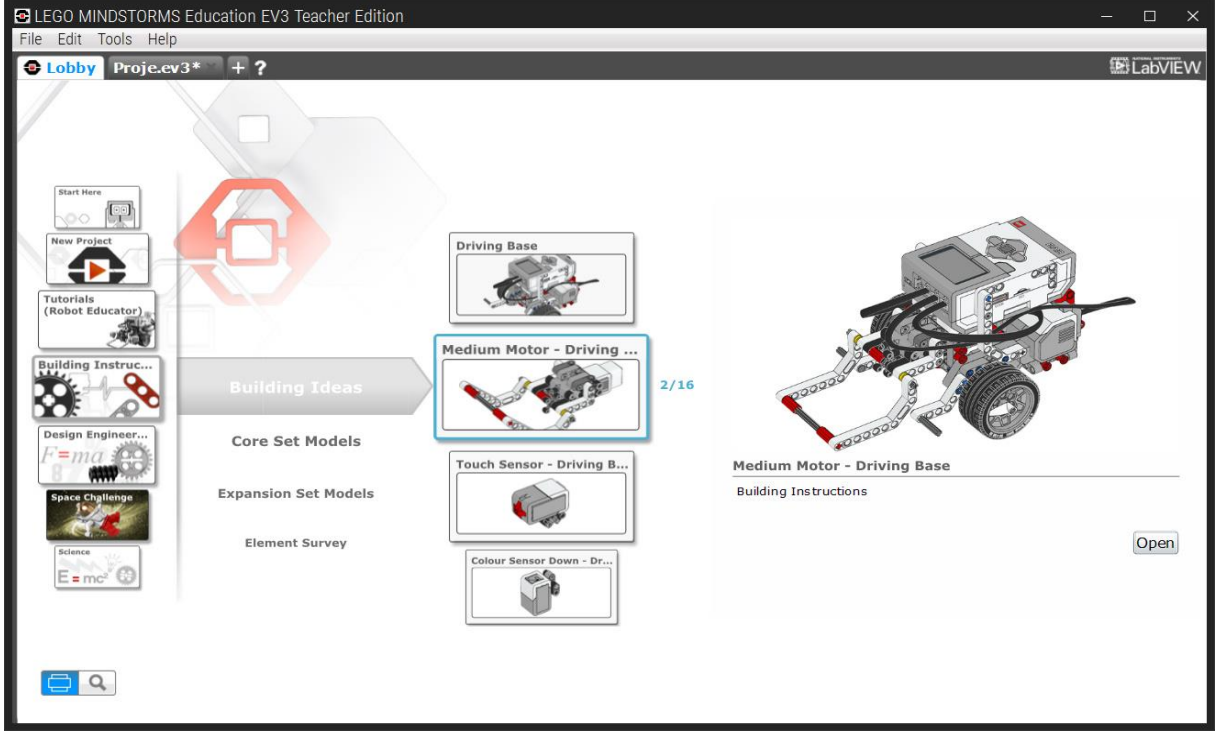
Büyük motor ve orta motor blokları sadece bir motorun dönmesini sağlamak için kullanılır. Kullanımı direksiyon ve palet hareketi blokları ile aynıdır.



Resim 14. Büyük Motor ve Orta Motor Blokları

1.4.6. Uygula: Orta Motor

Öğrencilerden orta motoru var olan temel tasarım robotuna eklemeleri istenir. EV3 yazılımının Lobby ekranında Building Instructions > Building Ideas > Medium Motor – Driving Base > Open adımları takip edilerek açılan yönergeye uygun olarak öğrenciler orta motoru robotlarına ekleyebilirler.



Resim 15. Orta Motorun Robota Eklenmesi

Öğrenciler belirtilen adımları izleyerek orta motora ile hareket eden kol tasarımını gerçekleştirdikten sonra rehber öğretmen, kolun orta motor kod bloğuyla aşağı ve yukarı nasıl hareket ettirebileceği gösterilir. Öğrencilerden benzer programı yaparak kolu farklı hızlarda hareket ettirmeleri istenir.

Not

Orta motor ile kolun, temel tasarım robotuna takılması için gerekli yönergeye aşağıdaki yollarla ulaşılabilir:

i) <https://le-www-live-s.legoocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-medium-motor-driving-base-e66e2fc0d917485ef1aa023e8358e7a7.pdf>

ii) EV3 yazılımı > Lobby > Building Instructions > Building Ideas > Medium Motor Driving Base

iii) Robot setiyle birlikte gelen kitapçığa bakılabilir.

2. TASARLA

Gün içerisinde robotun tekerinin çapını öğrenen öğrencilerden, bu bilgiyi kullanarak robotlarının 50 cm düz ilerleyip durmasını, sonra takılı olan kolu indirip LEGO parçaları ile yapılmış bir kutuyu (örnek bir LEGO cismi aşağıdaki resimde görülebilir) tutarak kendi etrafında 180 derece dönmesini ve başlangıç noktasına getirmesini sağlayan bir program tasarımları istenir. Tasarlama aşamasında öğrencilerin aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmeleri sağlanır.

Not

LEGO parçaları ile yapılacak kutu yönergesine aşağıdaki yollarla ulaşılabilir:

- i) <https://le-www-live-s.lego.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-cuboid-dc93b2e60bed2981e76b3bac9ea04558.pdf>
- ii) EV3 yazılımı > Lobby > Building Instructions > Building Ideas > Cuboid
- iii) Robot setiyle birlikte gelen kitapçığa bakılabilir.

Tanımlama: Öğrencilerden öncelikle robotun istenilen hareketleri yapabilmesi için neler gerektiğini belirlemeleri ve maddeler hâlinde yazmaları beklenir. Bu aşamada kâğıt ve kalem kullanarak programın algoritmasını veya akış diyagramını hazırlamaları istenir.

Örneğin robot;

- 50 cm ileri gidecek ve duracak,
- Kolunu kutunun üzerine indirecek,
- Kendi etrafında kutuyla birlikte dönecek,
- Başlangıç noktasına geri dönecek.

Fikir üretme: Bu aşamada öğrencilerin tanımlamada belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir:

- Öncelikle robotun hangi hareketleri yapacağı planlanmalıdır. Planlanan her bir hareketin tanımı yapılmalıdır (Örneğin, ilk harekette 50 cm ileri gidip durur ve etrafında 180 derece döner.). Bu aşamada öğrenciler hareketlerin programlanmasını hızlıca deneyebilirler.
- Robot kolunun indirilmesi için orta motor kullanılır.
- Kolun kutuyu kavraması sağlanır.
- Robotun kendi etrafında 180 derece dönmesi sağlanır.
- Robotun başlangıç noktasına geri gelmesi sağlanır.



Resim 16. Robot Kolunun Kutuyu (LEGO Parçasını) Taşımaları

Robotu planlanan şekilde hareket ettirmek için mat üzerindeki ilgili bölge kullanılabilir. Taşınacak kutu oluşturulurken setle gelen LEGO parçaları kullanılabilir. Kutunun taşınması ile ilgili oluşabilecek muhtemel sorunlarla ilgili öğrencilerden EV3 robot setinde bulunan parçaları kullanarak çözüm üretmeleri istenir (Eğitim programındaki bu ve benzeri bütün örnekler/programlar rehber öğretmenlere verilecektir).

3. ÜRET

Tasarla ve üret bölümlerinde öğrenciler aktif rol üstlenerek verilen problemi çözerler. Rehber öğretmen öğrencilere yalnızca zorlandıkları noktalarda destek olur. Öğrenciler bilgisayarda ve robot üzerinde çalışarak gerekli yazılım çözümlerini geliştirirler.

4. DEĞERLENDİR

Günün sonunda öğrencilerle halka oluşturulur ve aşağıdaki sorular üzerinden bir tartışma yürütülür:

- Robotun hızının değiştirilmesi ne gibi sonuçlara yol açtı? (Hız-zaman ilişkisi tartışılır)
- Palet hareketi (move tank) ile direksiyon hareketi (move steering) arasında fark var mıdır?
- Sizce teker büyüklüğünün bir önemi var mıdır?
- Programınızda tur sayısını değiştirmeden sadece teker büyüklüğünün değiştirilmesi ne gibi sonuçlara yol açar?

Bu soruların cevaplarına göre farklı robotların da tasarlanabileceğine dair örnekler verilebilir.

5. İLAVE ETKİNLİK

5.1. Sonsuz İşareti Şeklinde Hareket Etme

Bu ilave etkinlik zaman kalması durumunda yapılabilir. İki nesne (örneğin iki su şişesi) sınıf içinde müsait bir alana, aralarında 40-50 cm olacak şekilde yerleştirilir. Robotun bu nesnelere dokunmadan etraflarında “8” veya “sonsuz sembolü” çizerek hareket etmesi sağlanır. Robotun izleyeceği örnek yol aşağıdaki resimdeki gibi olabilir.



Resim 17. Örnek Yol

Etkinliği zenginleştirmek için iki nesne arasındaki mesafe artırılabilir ya da azaltılabilir. Ayrıca robotun başlangıç noktasına göre algoritma ve tasarım değişiklik gösterebilir. Programı oluşturmaya başlamadan önce grupların tasarlama adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme süreçlerini gerçekleştirmeleri gerekir (Rehber öğretmen kendisine verilen örnek çözümü inceleyebilir ama çözümü öğrencilerle paylaşmamalıdır, görevi öğrencilerin kendilerinin yapmasına izin vermelidir.).

5.2. Yarışma

Öğrencilerin daha eğlenceli bir şekilde çalışmasını sağlamak amacıyla bir yarışma düzenlenebilir. Yarışma grup olarak yapılır. Yarışmanın sadece eğlenmek için yapıldığı, sonucun başka bir şekilde değerlendirilmemesi gerektiği vurgulanır.

Öğrencilerden ön tekerlekleri matın kısa kenar çizgisi üzerinde bulunan bir robotun hareket ederek karşı kenar çizgisine yine ön tekerlekleri gelecek şekilde ilerlemesini sağlayacak bir program oluşturmaları istenir. Matın A yüzeyindeki 98 cm'lik kesikli kırmızı çizgi bu amaçla kullanılabilir. Robotun rota üzerinde kendi etrafında tam 180 derece dönmesi gerektiği unutulmamalıdır.

Amaç, robotun en kısa sürede görevi tam olarak (hatasız şekilde) gerçekleştirmesidir. Rehber öğretmen görevin tamamlanması için azami bir süre belirleyerek yarışma öncesinde öğrencilere duyurur. Grupların görevi tamamlama süreleri rehber öğretmen tarafından takip edilir ve en kısa sürede görevi tamamlayan grup yarışmayı kazanır. (Rehber öğretmen kendisine verilen örnek çözümü inceleyebilir ama çözümü öğrencilerle paylaşmamalıdır, öğrencilerin görevi kendilerinin yapmasına izin vermelidir.).

3. Hafta: Robotlarla Ses, Metin, Resim

Ön Bilgi:

- Öğrenciler robotik kavramını temel düzeyde bilir.
- Öğrenciler robot setiyle farklı robotik tasarımlar yapmıştır.
- Öğrenciler robot setini programlamak için grafik arayüzünü kullanmıştır.
- Öğrenciler açı, tur, çap (yarıçap), eşitlik ve denklem kavramını öğrenmiştir.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.

Haftanın Kazanımları:

- Öğrenciler akıllı tuğlanın çeşitli sesleri (dosya, frekans ve nota) çıkarması için gerekli programlama adımlarını oluşturur.
- Öğrenciler akıllı tuğla ekranının görüntüsünü (metin, şekil, imaj) düzenlemek için gerekli programlama adımlarını oluşturur.
- Öğrenciler robotu programlarken döngü mantığını kurup uygulayabilir.

Haftanın Amacı:

Bu haftanın amacı öğrencilerin öncelikle EV3 yazılımının “Eylem (Action)” sekmesinde bulunan “Ses (Sound)”, “Ekran (Display)” ve “Tuğla Durum Işığı (Brick Status Light)” bloklarını kavramalarını ve bu blokları programlama sırasında çeşitli amaçlar için kullanırken gerekli düzenlemeleri yapabilmelerini sağlamaktır.

Kullanılacak Malzemeler:

Robot seti, bilgisayar.

Haftanın İşlenişi:

Gözele: Eylem (Action) sekmesinde bulunan Ses (Sound), Ekran (Display) ve Tuğla Durum Işığı (Brick Status Light) bloklarının özelliklerini ve bu blokları kullanmak için gereken “döngü” kavramını inceleme.

Uygula: Her bir bloğun bileşenlerini programlayarak uygulama ve döngü kavramı ile örnek uygulama geliştirme (döngüsüz ve döngülü programlar oluşturma).

Tasarla: Robotun istenilen işlemleri yapabilmesi için gereken bileşenleri tanımlama ve planlama.

Üret: Robotun planlanan işlemleri yapabilmesi için gereken robot tasarımını ve programlama adımlarını oluşturma.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği.

1. GÖZLE VE UYGULA

1.1. Gözle: Ses (Sound) Bloğu

Öncelikle Ses (Sound) bloğu öğrencilere anlatılır ve bazı örnekler gösterilir.

Robot iki tür ses çalabilir: (i) “bip” sesi gibi basit bir ton, (ii) önceden kaydedilmiş “alkış” gibi bir ses veya “merhaba” gibi bir kelime. Programda bir ses bloğu kullanıldığında robot “konuşabileceği” için daha etkileşimli ve gerçekçi görünecektir. Programlama paletinden bir ses bloğu seçilir ve programlama alanına yerleştirilir. Blok yerleştirildikten sonra mod seçilir ve hangi ses duyulmak isteniyorsa ona göre ayarlar yapılır.

Ses bloğunun dört modu vardır:

- Play File: “Merhaba” gibi önceden kaydedilmiş bir sesi çalar.
- Play Tone: Belirli bir süre için belirli bir frekansta bir ton çalar.
- Play Note: Belirli bir süre için piyanodan bir nota çalar.
- Stop: Çalmakta olan tüm sesleri durdurur.

Yukarıdaki modlara göre aşağıdaki seçenekler düzenlenir.

File Name

Dosya Oynat (Play File) modunda, Dosya Adı (File Name) alanı tıklanarak menüden bir ses seçilebilir. Hayvanlar, renkler, iletişim ve sayılar gibi kategorilerden bir ses seçilebilir. Ayrıca, Araçlar (Tools) menüsünde yer alan Ses Düzenleyicisi (Sound Editor) kullanılarak ses kaydedilebilir ve eklenebilir.

Volume

Çalınmak istenen sesin seviyesini ayarlamak için “0” (yumuşak) ile “100” (yüksek) arasında bir sayı girilebilir.

Play Type

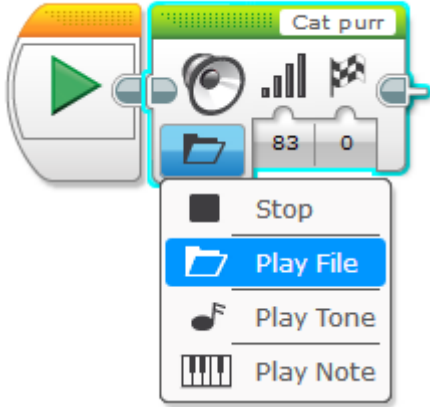
Ses çalmaya başladığında ne olacağını kontrol etmek için Çalma Türü (Play Type) ayarı kullanılabilir. Ses çalmayı durdurana kadar programı duraklatmak için tamamlanmasını bekle (0) ögesi seçilebilir. Ses çalarken programın bir sonraki bloğu çalıştırmaya devam etmesi için bir kez çal (1) ögesi seçilebilir. Tekrarla (2) seçilirse, program kalan blokları çalıştırırken ses tekrar eder. Çoğu program için tamamlanmasını bekle (0) seçeneği seçilecektir.

Note - Tone

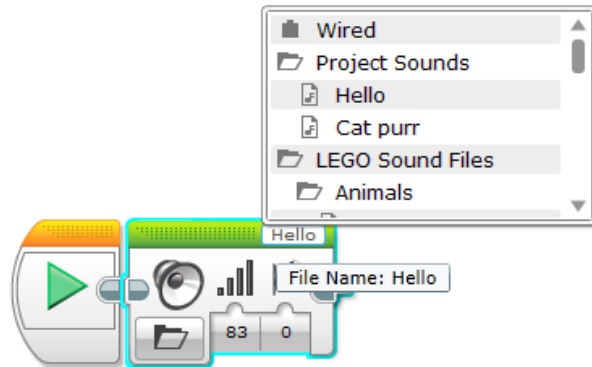
Tercih edilen moda bağlı olarak, piyano klavyesinden bir nota veya hertz (Hz) cinsinden bir ton (ton) seçilebilir. İnsan kulağı 440 Hz tonu (varsayılan frekans değeri) net bir şekilde duyar. Bu ton robota etkileşim katmak için kullanılabilir gibi oluşturulan programları test etmek için de kullanılabilir. Örneğin belirli bir programlama bloğunun çalışmasının tamamlandığını göstermek için ses çalınabilir.

Duration

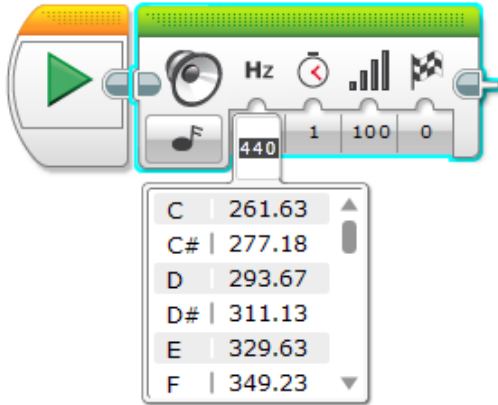
Süre (Duration) kutusuna notanın veya zil sesinin çalması istenen süresi saniye cinsinden girilebilir.



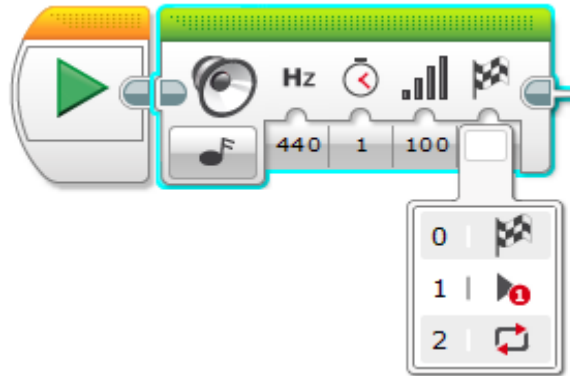
Resim 18. Play File Adım 1



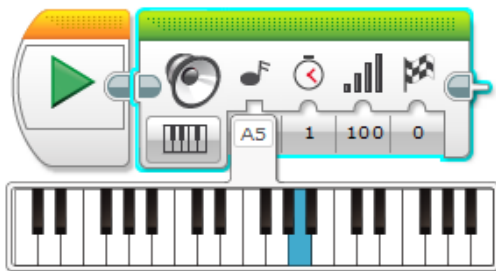
Resim 19. Play File Adım 2



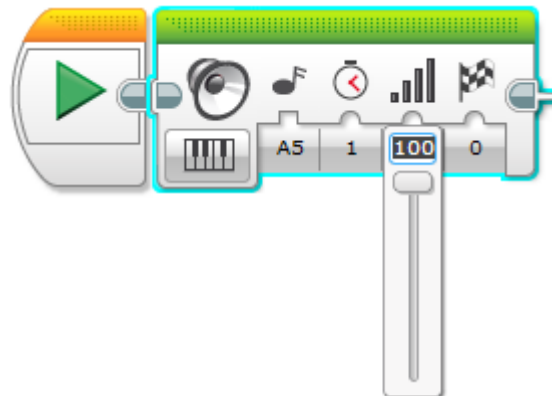
Resim 20. Play Tone Adım 1



Resim 21. Play Tone Adım 2

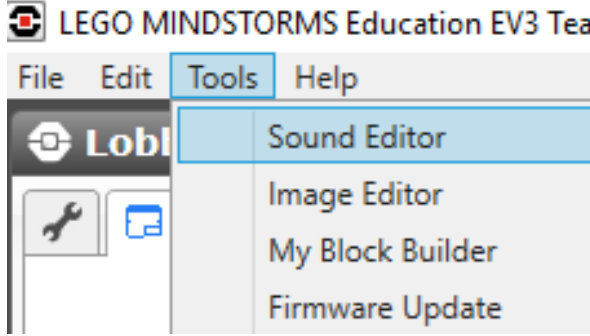


Resim 22. Play Note Adım 1

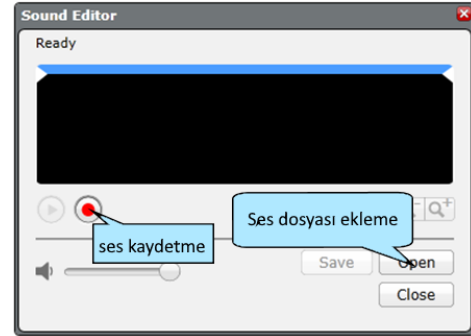


Resim 23. Play Note Adım 2

Kişisel ses dosyası oluşturmak veya ses dosyası eklemek için EV3 yazılımında Araçlar (Tools) menüsünden Ses Düzenleyicisi (Sound Editor) seçilir. Açılan pencereden kayıt düğmesine tıklanılarak ses kaydı yapılır. Bir isim verilerek kaydedilen ses dosyasına, Ses (Sound) bloğunun Dosya Oynat (Play File) kısmından ulaşılabilir.



Resim 24. Ses Kayıt Adım 1



Resim 25. Ses Kayıt Adım 2

1.2. Uygula: Ses (Sound) Bloğu

Nota çalma: Öğrencilerden ardışık dört notayı çalmaları istenir. Örnek program aşağıdaki resimde görülmektedir.



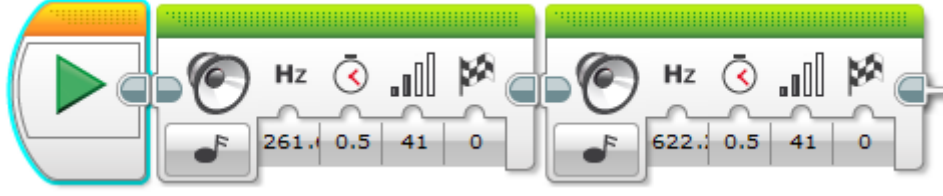
Resim 26. Örnek Program

Kendi hoş geldin mesajını oluşturma: Öğrencilerden kendi hoş geldin mesajlarını oluşturup bu mesajı robota söyleten bir program oluşturmaları istenir.

1.3. Gözle: Siren Sesi

Öğrencilere daha önce siren sesi duyup duymadıkları sorulur. Öğrenciler sadece polis, itfaiye ve ambulans sirenlerinden bahsederlerse AFAD'ın sitesindeki <https://www.afad.gov.tr/ikaz-alarm-isaretleri> ikaz alarm seslerinden de bahsedilir. Verilen örneklerdeki araçların (ambulans, polis arabası vb.) siren seslerinin ortak özelliği öğrencilere sorulur. Eğer uygun cevap veremezlerse siren seslerinin tekrarlayan daha kısa seslerden oluştuğu vurgulanır.

Robotun programlanmasıyla siren veya ikaz sesleri çıkarabileceğinden bahsedilir. Aşağıdaki resimde görülen program ekranda gösterilir. Program robota yüklenir ve çalıştırılır. Robotun iki farklı frekansı 0,5 saniye süreyle ardışık olarak çaldığı açıklanır.



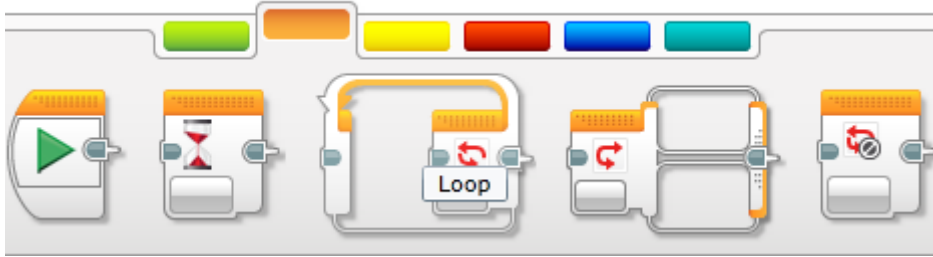
Resim 27. Örnek Program

Rehber öğretmen, 4 saniyelik siren sesi için robotun her iki frekansı sıra ile dörder defa çalması ve programın da buna göre düzenlenmesi gerektiğinden bahseder. Aşağıdaki resimde örnek program görülmektedir.



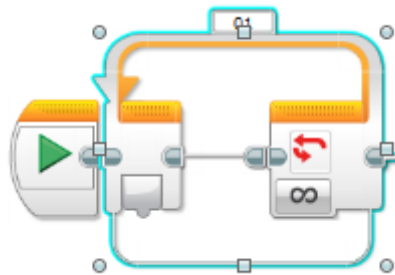
Resim 28. Örnek Program

Rehber öğretmen, tekrarlanan işlemler için Akış Kontrol (Flow Control) sekmesinden Döngü (Loop) bloğunun seçilmesi gerektiğini söyler.



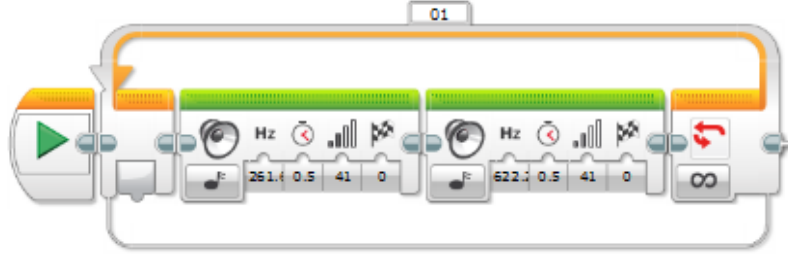
Resim 29. Flow Control Sekmesi

Daha sonra, Döngü bloğunun başla komutuna eklenmesi gerektiği ifade edilir. Öğrenciler ilk defa döngü kavramıyla karşılaştığı için programlama sürecinde kullanılan döngülerle ilgili kısa bilgiler paylaşılır. Özellikle döngülerin tekrarlı adımlar yapılırken kullanıldığı söylenir.



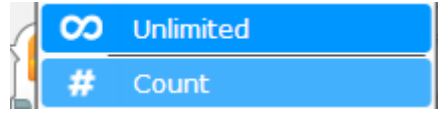
Resim 30. Döngü (Loop) Bloğu

Döngü (Loop) bloğunun içerisine tekrar etmesi istenilen iki frekans eklenir.

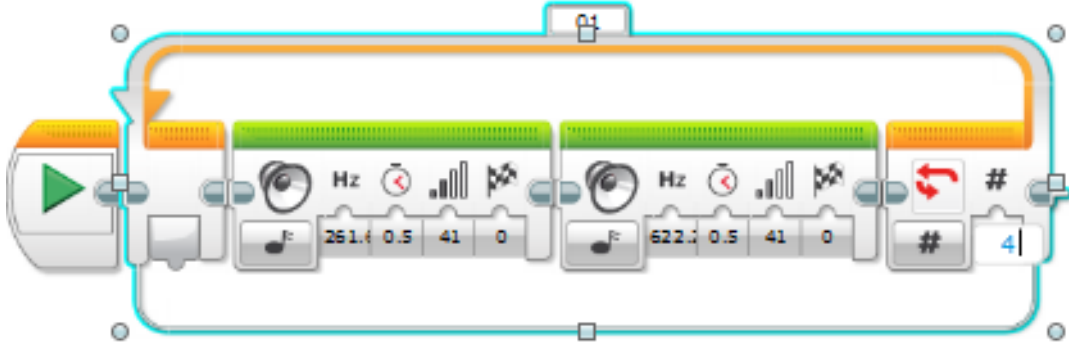


Resim 31. Örnek Program

Program robota yüklenir ve öğrencilere iki sesin sürekli olarak ardı ardına çaldığı gösterilir. Eğer her iki sesin ardışık olarak dörder defa çalması arzu edilirse döngüdeki Sınırsız (Unlimited) simgesine tıklanıp Sayma (Count) seçeneğinin seçilmesi ve değerinin 4 yapılması gerekir.



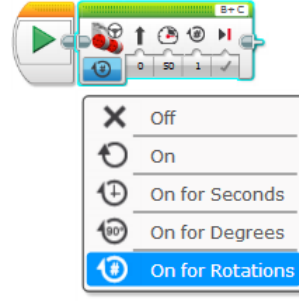
Resim 32. Adım Sayısının Belirlenmesi 1



Resim 33. Adım Sayısının Belirlenmesi 2

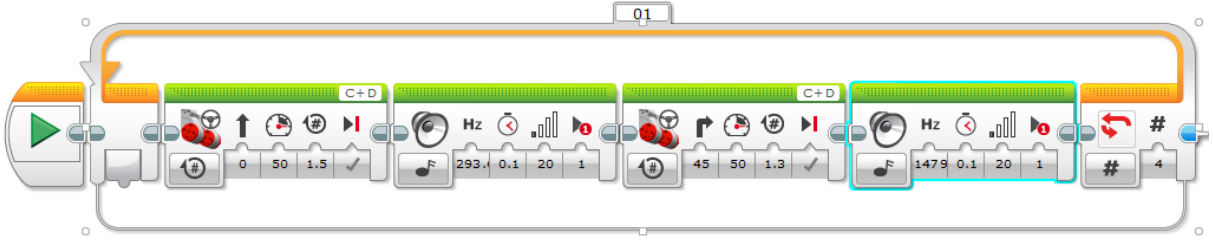
1.4. Uygula: Kare Çizen Robot

Öğrencilerden robotun 20 cm'lik bir kare üzerinde ilerlemesi için bir program oluşturmaları istenir. Programı oluştururken Döngü (Loop) bloğunu kullanmaları; robotun 20 cm ilerleyince bir ses çıkarmasını, olduğu yerde sağa veya sola (90 derece) dönünce de farklı bir ses çıkarmasını sağlamaları istenir. (Öğrenciler robotun 20 cm ilerleyebilmesi ve sağa veya sola (90 derece) dönebilmesi için gerekli bilgileri önceki derste öğrenmişlerdir). 20 cm'yi ayarlayabilmek için "On for Degrees" veya "On for Rotations" seçeneklerinden birinin kullanılmasının gerektiği hatırlatılmalıdır.



Resim 34. Dönüş Şeklinin Belirlenmesi

Aşağıdaki resimde programın tamamlanmış hâline örnek verilmiştir (Dönüş sayısı için verilen 1,3 değeri robota ve fiziksel ortama göre değişiklik gösterebilir):



Resim 35. Örnek Program

1.5. Gözle: Ekran (Display) Bloğu

Robotu hareket ettirme ve ses çalmanın yanı sıra bir EV3 programı “akıllı tuğla”nın ekranını kontrol edebilir. Ekran 178 piksel genişliğinde ve 128 piksel yüksekliğindedir. (Piksel, ekranda görüntüyü oluşturan küçük noktalardır.)



Resim 36. EV3 Ekranı

Rehber öğretmen öğrencilere yukarıdaki resmi gösterir.

- Ekranın sol üst köşesinden sağ üst köşesine kadar 178 nokta olduğunu ve herhangi bir noktanın ekranın sol kenarından uzaklığını belirtmek için Ekran (Display) bloğunda “x” ifadesinin kullanıldığını söyler.

- Ekranın sol üst köşesinden sol alt köşesine kadar 128 nokta olduğunu ve herhangi bir noktanın ekranın üst kenarından uzaklığını belirtmek için Ekran (Display) bloğunda “y” ifadesinin kullanıldığını söyler.
- Ekranda herhangi bir noktayı belirtmek için de noktanın ekranın sol kenardan kaç nokta uzakta olduğunu (yani x değeri) ve noktanın ekranın üst kenarından kaç nokta uzakta olduğunu (yani y değeri) bilmeleri gerektiğini söyler.

Display bloğuyla oynamak eğlenceli olduğu kadar programı test etmenin etkili yollarından biridir. Örneğin sensörün düzgün çalışıp çalışmadığını görmek için ekranda bir sensör ölçümü görüntülenebilir.

Akıllı tuğla ekranında bir görüntüyü (gülen yüz), metni (“Merhaba!”) veya bir şekli (dolu bir daire gibi) görüntülemek için Ekran (Display) bloğu kullanılır. Tek bir Ekran bloğu ekrana bir defada birden fazla görüntü veya metin satırı koyamaz, bu nedenle ekranı oluşturmak için bazen bir dizi Ekran bloğunun kullanılması gerekebilir.

Ekran (Display) Bloğu Ayarları

Bir ekran bloğu “akıllı tuğla” ekranına bir görüntü eklediğinde, program bir sonraki bloğa geçer (örneğin bir move bloğu). Akıllı tuğla ekranı, başka bir ekran görüntülemek için yeni bir ekran bloğu kullanılana kadar görüntüyü göstermeye devam eder. Bir program sona erdiğinde akıllı tuğla hemen menüye döner. Bu, programdaki son blok bir Ekran bloğu ise program sona ereceği için ekranda ne olduğunu görmek için zaman kalmayacağı anlamına gelir. Ekranda ne olduğunu görmek ve programın hemen sona ermesini engellemek için Bekle (Wait) bloğu eklenir.

- Image: Ekranda mutlu bir yüz gibi seçilmiş bir resim gösterir.
- Shapes: Ekranda çizgi, daire, dikdörtgen veya nokta gösterir.
- Text: Ekranda bir metin satırı gösterir.
- Reset Screen: Ekranı temizler ve ekran blokları olmayan bir program çalıştırıldığında normalde görülen “MINDSTORMS” logosunu gösterir.

Alt Modlar (Sub Modes)

Bazı modlarda alt seçenekler vardır. Ekran bloğunda şekiller modu seçilirken dört alt moddan birinin tercih edilmesi gerekir (çizgi, daire, dikdörtgen ve nokta). Daire modu, ekran bloğunun EV3 ekranında bir daire şekli göstermesini sağlar. Dairenin konumunu, yarıçapını, dolgusunu ve rengini yapılandırmak için bloktaki ayarlar kullanılabilir.

Dosya İsmi (File Name)

Görüntü modundayken gözler, ifadeler, nesnelere ve LEGO gibi kategorilerden bir görüntü seçmek için Dosya Adı (File Name) alanı kullanılır. Araçlar (Tools) sekmesinden Resim düzenleyicisine gidilerek kişisel resimler oluşturulabilir veya yüklenebilir.

Ekranı Temizle (Clear Screen)

Ekranı Temizle (Clear Screen) ayarı, yeni bir şey göstermeden önce ekranı boşaltmayı (doğru olarak ayarlandığında) veya ekranda zaten olanlara (yanlış olarak ayarlandığında) yeni bir şey eklemeyi seçmeyi sağlar. Ekranda birden fazla nesne göstermek için bir dizi ekran bloğuna ihtiyaç olacaktır. İlk blok yeni bir şey göstermeden önce ekran temizlenmeli ve diğer bloklar ekrana bir şey eklemelidir.

Bunu başarmak için ilk blokta Clear Screen ayarını “true (doğru)” yapmak ve izleyen bloklarda “false (yanlış)” olarak ayarlamak gerekir.

Yarıçap ve Doldurma (Radius and Fill)

Bazı ayarlar, ekran bloğunun farklı modlarına özgüdür. Örneğin “radius (yarıçap)” ayarı bir dairenin boyutunu belirtir ve “fill (doldurma)”, düz bir daire yapılmasına (doğru) veya yalnızca anahat çizilmesine (yanlış) izin verir.

Renk (Color)

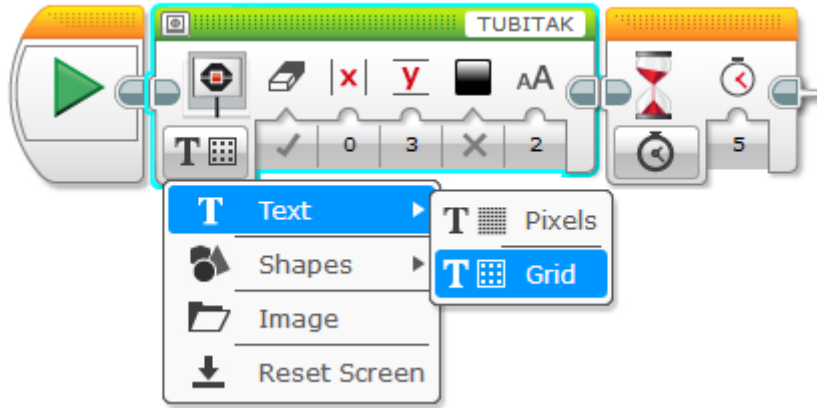
Renk (Color) ayarı siyah (yanlış) olarak ayarlanmak istenirse ve daha önce bir alan siyah bir daireyle doldurulduysa, renkler beyaz (doğru) olarak ayarlanarak üstüne metin eklenebilir.

Metin ve Yazı Boyutu (Text and Font Size)

Metin (Text) modunda, metin alanına, “MINDSTORMS” gibi, görüntülenmek istenen bir metin satırı girilir. Metin satırı sayılar içerebilir ve “font size (yazı tipi boyutu)” değeri “0” (küçük), “1” (kalın) veya “2” (büyük) olarak ayarlanarak metnin boyutu değiştirilebilir.

1.5.1. Gözle: Metin (Text)

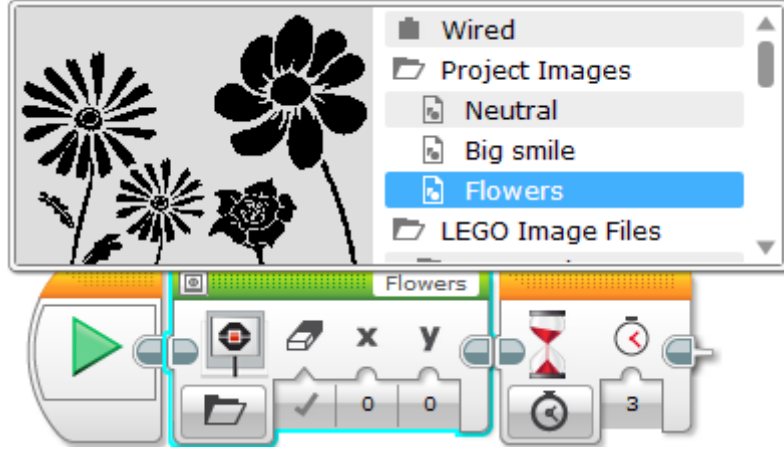
Text seçeneği ile ekrana yazı (örneğin TUBITAK) yazılabilir fakat “wait (bekle)” eklenmezse yazı akıllı tuğla ekranında görünmez. Ekranda gösterilirken hata yaşanabileceği için Türkçe karakterler kullanılmamalıdır. Aşağıdaki resimde display bloğu örneğinde “x” değeri “0” (sıfır) y değeri “3” olarak girilmiştir. TUBITAK yazısı ekranın hemen sol kenarından (Çünkü “x” değeri sıfır girilmiştir.) ve ekranın üstünden 3 pixel boşluk bırakılarak (Çünkü “y” değeri 3 girilmiştir.) yazılacaktır.



Resim 37. Ekran Yazısı Yazdırma

1.5.2. Gözle: Resim (Image) Dosyaları

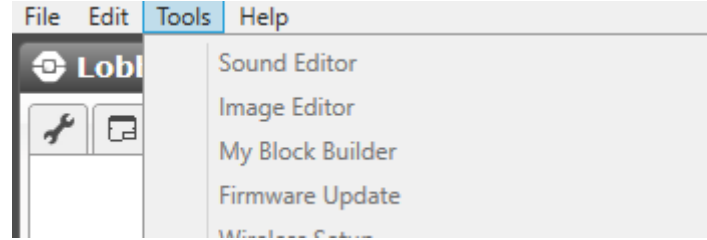
Hazır resim dosyaları eklenebilir, benzer şekilde “wait (bekle)” eklenmezse akıllı tuğla ekranına görüntü gelmez. Aşağıdaki resimde display bloğu örneğinde “x” değeri 0 (sıfır), “y” değeri 0 (sıfır) olarak girilmiştir. Flowers dosyası ekranın sol üst köşesinden itibaren (Çünkü “x” ve “y” değeri sıfır girilmiştir.) görüntülenecektir.



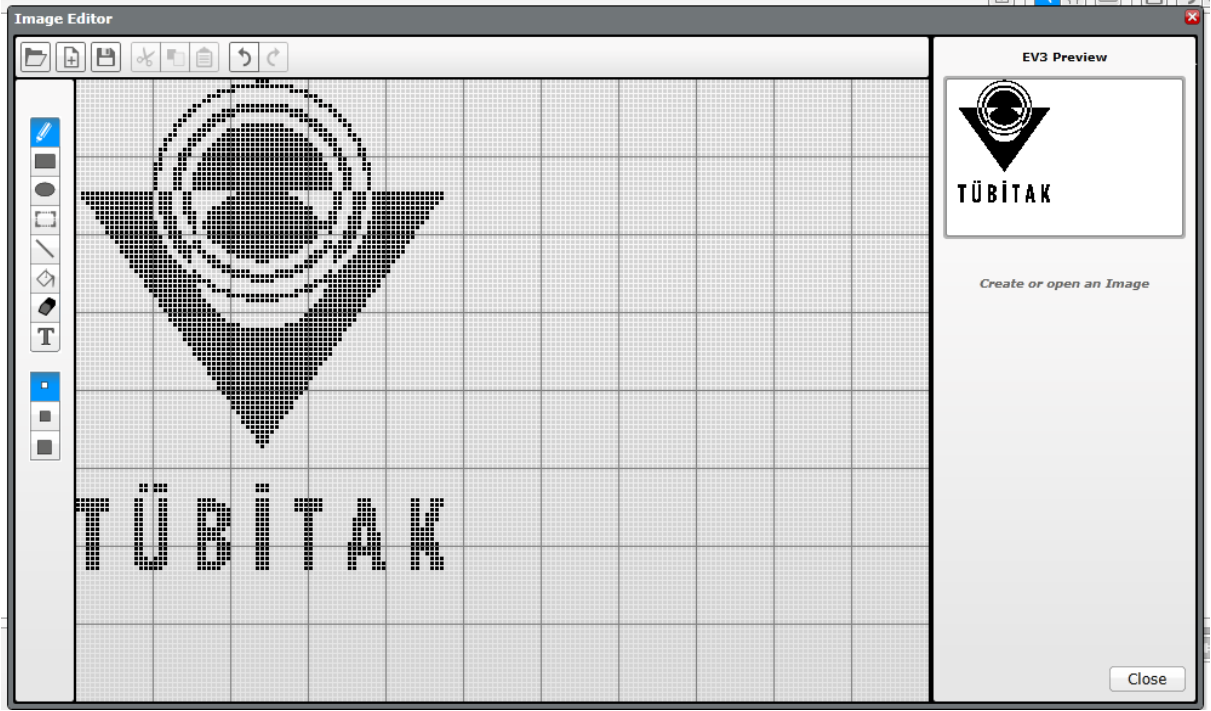
Resim 38. Ekran Resim Basma

1.5.2. Gözle: TÜBİTAK Logosunu Yükleme

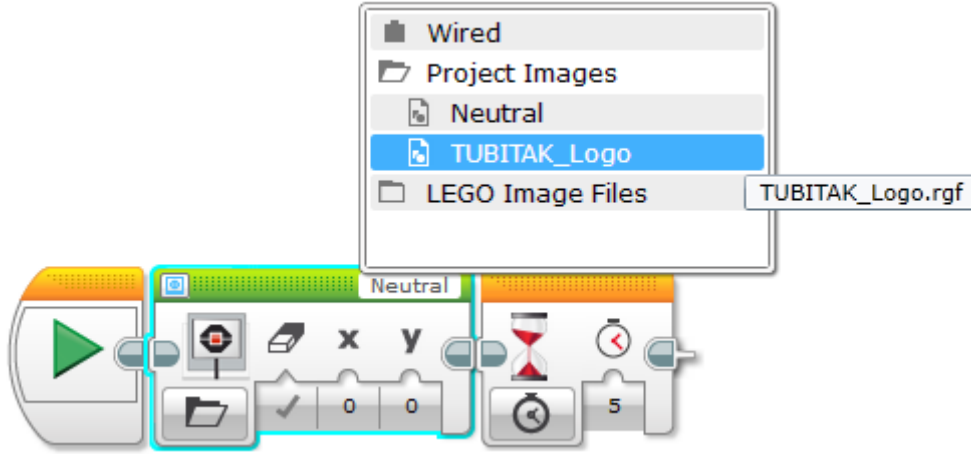
TÜBİTAK logosunu yüklemek için gerekli adımlar aşağıdaki resimlerde görülebilir. TÜBİTAK logosunu yüklemek için EV3 yazılımında Araçlar (Tools) menüsünden Resim Editörü (Image Editor) seçilir. Açılan pencereden Aç (Open) düğmesine tıklanılarak resim yüklenmeye başlanır. Bir isim verilerek kaydedilen resim dosyasına Ekran (Display) bloğunun Project Images kısmından ulaşılabilir.



Resim 39. Logo Ekleme Adım 1



Resim 40. Logo Ekleme Adım 2

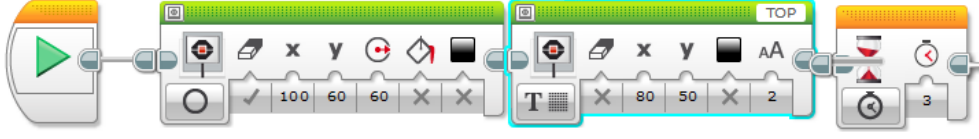


Resim 41. Logo Ekleme Adım 3

1.6. Uygula: Ekran (Display) Bloğu

1.6.1. Metin Yazdırma

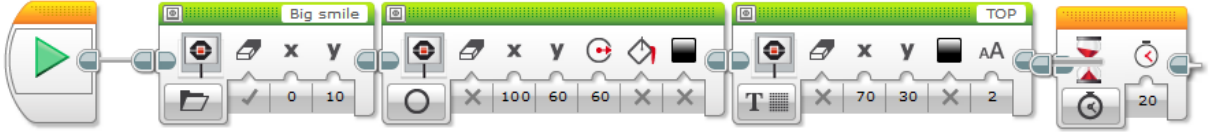
Öğrencilerin bir daire içine isimlerini veya soy isimlerini yazdırabilecekleri bir uygulama yapılır. Aşağıdaki resimde ilk "display" bloğunda dairenin merkezi için "x" değeri 100 ve "y" değeri 60 girilmiştir. Yarıçap olarak da 60 girilmiştir. Bu daire tanımlaması ile merkezi ekranın sol kenarından 100 pixel ve ekranın üst kenarından 60 pixel uzaklıkta olan 60 pixel yarıçapında bir daire çizilecektir.



Resim 42. Örnek Program

1.6.2. Resim (Image) Gösterme

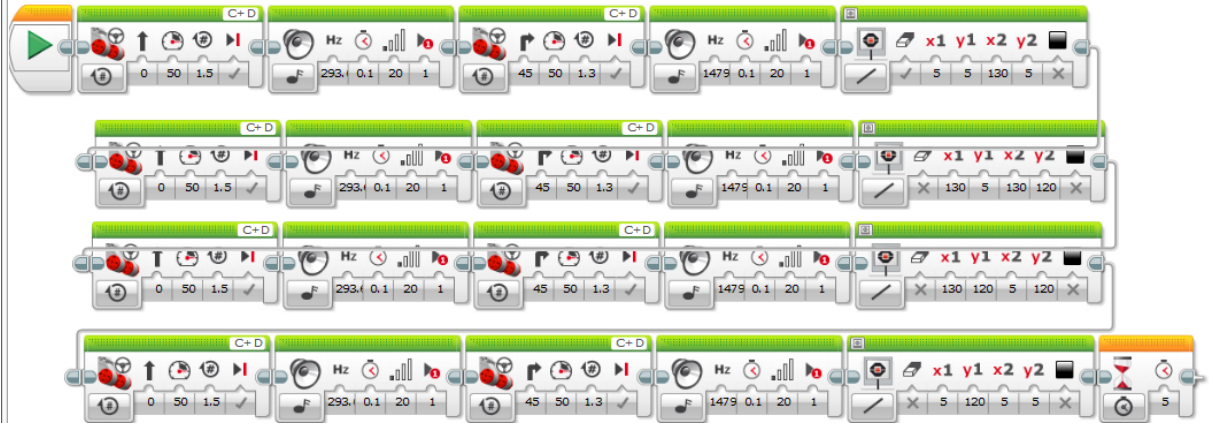
Öğrencilerin bir daire içine “big smile” resmi ile isimlerini veya soy isimlerini yazdırabilecekleri bir uygulama yapılır. Örnek program aşağıdaki resimde görülmektedir.



Resim 43. Örnek Program

1.6.3. Şekil Oluşturma

Bu aşamada, 1.4. Uygula başlığında yapılan kare çizen robot örneği tekrar düzenlenir. Her çizgi tamamlandığında akıllı tuğlanın ekranında o çizgi de gösterilir. Kare tamamlanınca ekranda 5 saniye kalması sağlanır.



Resim 44. Örnek Program

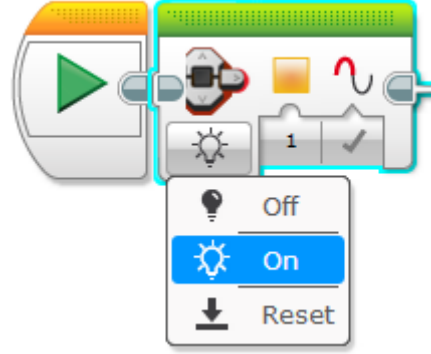
1.7. Gözle: Tuğla Durum Işığı (Brick Status Light) Bloğu

Öncelikle Eylem (Action) sekmesinde bulunan Tuğla Durum Işığı (Brick Status Light) bloğu programa eklenerek başlanır.

EV3 düğmelerini çevreleyen Tuğla Durum Işığı (Brick Status Light) normalde yeşildir. Bir program çalışırken yanıp söner. Tuğla Durum Işığı bloğu ile ışığın nasıl yanacağı seçilebilir. Seçilebilecek üç mod aşağıdaki gibidir:

- On: Işığı açar ve bir renk seçilmesini sağlar: yeşil (0), turuncu (1), kırmızı (2). Darbe ayarı ile ışığın yanıp söneceği mi (doğru) yoksa sürekli açık mı kalacağı (yanlış) seçilebilir.

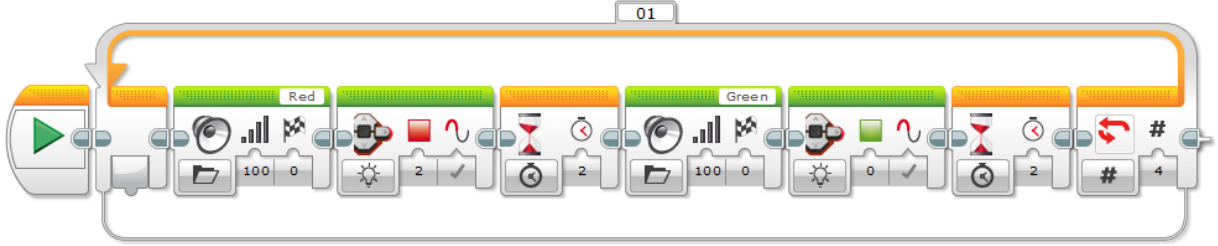
- Off: Işığı kapatır.
- Reset: Tuğla Işığı (Brick Status) bloğu ile herhangi bir işlem yapılmadan program çalıştırıldığında yanıp sönen yeşil ışığın gösterilmesini sağlar.



Resim 45. Tuğla Durumu Işığı Ayarları

1.8. Uygula: Tuğla Durum Işığı (Brick Status Light) Bloğu

Bu uygulamada sırası ile kırmızı ve yeşil ışığın yakıldığı bir program hazırlanır. Her bir ifade arasında iki saniye beklenir ve bu işlem beş defa tekrar edilir. Örnek program aşağıdaki resimde görülmektedir.



Resim 46. Örnek Program

2. TASARLA

2.1. Dans Eden Robot

Tasarla aşamasında öğrencilerden robotlarını dans edecek şekilde programlamaları istenir. Robot dans ederken hareketler, müzik ve akıllı tuğla ekranındaki görseller senkronize bir şekilde kullanılmalıdır.

Öğrencilerden dans eden robotun kodunun nasıl yazılacağı üzerinde düşünmeleri istenir. Öğrenciler grup olarak tartışır. Gerekli noktada rehber öğretmen onlara yardımcı olabilir. Fakat öğrencilere tam bir çözüm verilmemelidir. Gruplar çözümü kendileri üretmelidir, tam bir çözümün verilmesi öğrencilerin yaratıcılıklarını olumsuz yönde etkileyebileceğinden tavsiye edilmez. Dans eden robotu tasarlamak için öğrencilerin aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmeleri gerekir.

Tanımlama: Öncelikle öğrenciler robotun dans edebilmesi için neler gerektiğini belirlemelidir. Bu bir dans stili, farklı şekillerde hareket eden, nota çalan ve ekranda uygun ifade gösteren bir robot olabilir. Öğrenciler gerekli işlemleri maddeler hâlinde yazmalıdır.

Örneğin;

- Nota çalacak,
- Nota ile paralel olarak dans edecek (sağ, sol, ileri, sağ, sol, geri hareket edecek),
- Akıllı tuğla ekranında göz hareketleri ile sağa sola dönüş ifadesi; ileri gidince mutlu ifade, geri gidince mutsuz ifade gösterecek,
- Dansı sonlandıracak.

Fikir üretme: Bu aşamada öğrencilerin, tanımlamada belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir:

- Öncelikle robotun hangi hareketleri yapacağı planlanmalıdır. Planlanan her bir hareketin tanımı yapılmalıdır (Örneğin robot önce 0,5 saniye sağa doğru döner ve durur. Sonra iki katı hızlı bir şekilde yine 0,5 saniye sağa doğru döner ve durur.). Bu aşamada öğrenciler hareketlerin programlanmasını hızlıca deneyebilirler.
- Sonra her bir harekette tuğlanın çıkaracağı ses (dosya, frekans veya nota) belirlenir. Öğrencilerin internetten müzik dosyaları indirmelerine veya kendi müziklerini programlamalarına izin verilmelidir.
- Hareket ve ses ile tuğla ekranında hangi görsellerin gösterileceği de belirlenmelidir.
- Bu sürecin sonunda öğrenciler dans hareketlerini, müziğini, tuğla görüntülerini ve durum ışığı bilgilerini birlikte anlatabilmelidir.

3. ÜRET

3.1. Dans Eden Robot

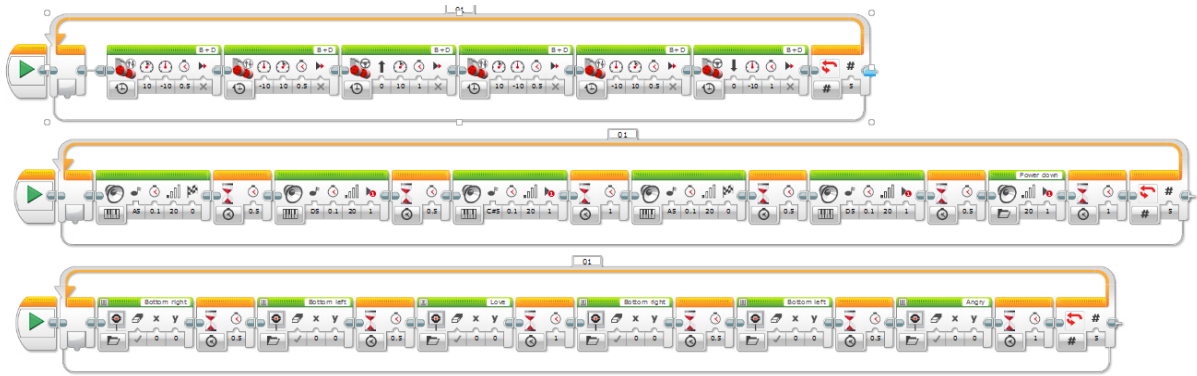
Bu bölümde de öğrenciler aktif rol üstlenir. Rehber öğretmen yalnızca yönlendirir ve öğrencilere takıldıkları noktalarda destek olur. Yani rehber öğretmen yalnızca ihtiyaç anında destek sağlamalıdır. Üret aşamasında, öğrencilerden bir önceki adımda tasarladıkları dans eden robot planını kullanarak probleme daha yapılandırılmış bir çözüm önerisi geliştirmeleri istenir. Öğrenciler bilgisayar ve robot başında çalışarak gerekli yazılım çözümlerini geliştirirler. Burada öğrencilerin en çok zorlanacağı konulardan biri ses, hareket ve tuğla ekranı görüntüsü senkronizasyonunun sağlanmasıdır.

Püf noktalar (Kendilerinin keşfetmesine veya sorarak öğrenmelerine imkân vermek için bunları başlangıçta öğrencilere söylemek uygun değildir.):

- Hareketin süresi ile resmin (image) ekranda bekleme süresi aynı olmalıdır.
- Notanın çalma süresinden bağımsız olarak, bekleme süresi hareketin süresi ile aynı olmalıdır. Örneğin nota 0,1 saniye çalışırsa ve hareket 0,5 saniye

sürüyorsa, bir sonraki hareketin notasını oynatmak için hareket süresi kadar (0,5 saniye) beklemek gerekir.

Aşağıdaki resimde nota çalıp dans eden (sağ, sol, ileri, sağ, sol, geri hareket eden) ve göz hareketleri ile sağa veya sola dönüşü beraber yapan, ileri gidince mutlu, geri gidince mutsuz ifade gösteren robotu programlamak için gerekli kodlar verilmiştir. Bu kod blokları rehber öğretmen için örnek bir dans stildir. Öğrenciler farklı şekillerde hareket eden, nota çalan ve akıllı tuğla ekranında uygun ifade gösteren programlar yapabilir (Rehber öğretmen kendisine verilen örnek çözümü inceleyebilir ama çözümü öğrencilerle paylaşmamalıdır, öğrencilerin görevi kendilerinin yapmasına izin vermelidir).



Resim 47. Örnek Program

4. DEĞERLENDİR

Bu aşamada hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşüncelerini sağlamaktır. Bu sayede öğrenciler; problem çözme yetenekleri, dersin konusu ve kendileri ile ilgili gözlemler yaparak yeni bilgiler öğrenme, kendilerini değerlendirme ve sonraki çalışmalarını planlama açısından fırsatlar elde edecektir. Öğrencilerin şu soruları cevaplamaları istenebilir:

- Size bugün Tasarla ve Üret adımında verilen problemi nasıl tanımlarsınız? (problemi kendi cümleleri ile ifade etme).
- En çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
 - Dans adımlarına nasıl karar verdiniz, bunları programlarken ne tür sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
 - Müzik, tuğla ekranı ve tuğla ışığına nasıl karar verdiniz; bunları programlarken ne tür sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
- Kullandığınız yöntemler bu sıkıntıları gidermede başarılı oldu mu?

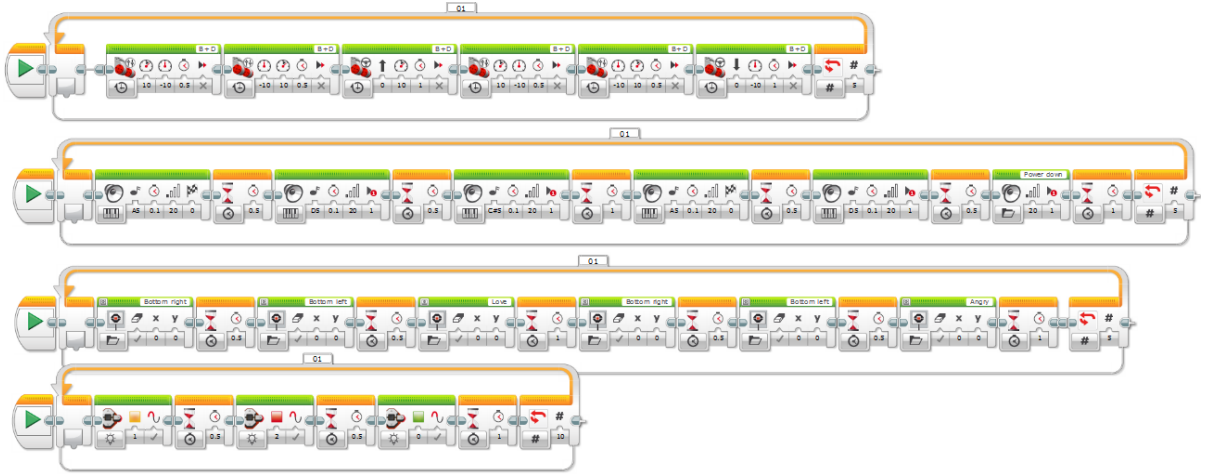
- Grup arkadaşınızla fikir ayrılıkları yaşadınız mı? Bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne/neler öğrendiniz?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilebilir.

5. İLAVE ETKİNLİK

5.1. Dans Eden Işıklı Robot

Bu ilave etkinlikteki amaç farklı şekillerde hareket eden, nota çalan, farklı şekillerde ışık yakan ve ekranında uygun ifade gösteren robotu programlamaktır. Bu program yukarıdaki etkinliğin devamı şeklinde yapılacaktır. Aşağıdaki resimde nota çalıp dans eden (sağ, sol, ileri, sağ, sol, geri hareket eden), ışık yakıp söndüren ve göz hareketleri ile sağa veya sola dönüşü aynı anda yapan, ileri gidince mutlu ifade, geri gidince ekranında mutsuz ifade gösteren robotu programlamak için gerekli kodlar verilmiştir. Bu kod blokları rehber öğretmen için örnek bir dans stildir (Bu ve benzeri programların kodları rehber öğretmenlerle paylaşılacaktır).



Resim 48. Örnek Program

4. Hafta: Dokunma Sensörü

Ön Bilgi:

- Öğrenciler robot setiyle farklı robotik tasarımlar yapmıştır.
- Öğrenciler robot setini programlamak için grafik arayüzünü kullanmıştır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler akıllı tuğla ekranını düzenlemek ve tuğlanın çeşitli sesler çıkarmasını sağlamak için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler robotu programlarken döngü mantığını kurup uygulayabilir.

Haftanın Kazanımları:

- Öğrenciler sensörleri tanır.
- Öğrenciler dokunma sensörünün çalışma mantığını ve kullanma yöntemlerini bilir.
- Öğrenciler dokunma sensörünün kullanıldığı farklı uygulamaların programlama adımlarını oluşturur.
- Öğrenciler bir nesneye çarptıktan sonra robotun yeni bir hareket yapması için gerekli programlama adımlarını oluşturur.

Haftanın Amacı:

Bu haftaya başlarken duyu organlarımız ile analogi yapılarak sensör kavramına giriş yapılır. Robot setiyle birlikte gelen sensörler (dokunma, mesafe, açı ve renk gibi) kısaca tanıtılır. Daha sonra öğrencilerin özellikle dokunma sensörünü verilen problemlerin çözümünde kullanması sağlanır.

Kullanılacak Malzemeler:

Robot seti, sensörler, bilgisayar.

Haftanın İşlenişi:

Gözle: Dokunma sensörünü kullanarak robotun harekete başlamasını sağlama ya da hareket hâlindeki bir robotu durdurma.

Uygula: Dokunma sensörünü kullanarak farklı uygulamalar geliştirme ve her bir bloğun çeşitli bileşenlerini programlayarak uygulama.

Tasarla: Dokunma sensörünü kullanarak robotun düz gitmesini, engele çarptığında ses çalmasını ve ekran görüntüsünün değişmesini sağlamak için gerekli bileşenleri tanımlama ve planlama.

Üret: Verilen görevlere göre robotu programlama.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği

1. GÖZLE VE UYGULA

1.1. Gözle: Sensör Nedir?

Robotların en önemli özelliklerinden biri algılamadır. Sensörler de bu amaçla kullanılan algılayıcılardır. Başka bir ifadeyle robotlar çevreden veri toplayan, bu veriyi işlemcileri ile işleyen, eyleme karar veren ve eylemi gerçekleştiren makinelerdir. Robotlar çevrelerindeki dünyayı algılamak için sensörlerini kullanırlar.

Rehber öğretmen sensörleri basitçe tanımladıktan sonra robotların sensörleri neden kullanması gerektiğini açıklamak adına: “Hareket eden bir robotun önündeki engelden kaçabilmesi için etrafındaki cisimleri algılayabilen bir sensöre ihtiyacı vardır.” gibi farklı örnekler verebilir. Ayrıca konunun daha iyi anlaşılmasını sağlamak için insanların duyularıyla robotların sensörlerini birbiriyle ilişkilendirebilir. İnsanların görme, işitme, dokunma, koku ve tat alma gibi duyularını çeşitli amaçlarla kullanması gibi robotlar da duyuları elektronik sisteme aktarabilmek ve bunların sonucunda bir işlemi gerçekleştirmek için sensörlerini kullanılır. Yani “Robotların duyu organları sensörleridir.” denilebilir.

Konuyu pekiştirmek için robotlarda sensörlerin kullanımına yönelik aşağıdaki linkte önerilen örnek videolar izlenebilir:

- Google's 5 Bets on the Future: Robots and Sensors:
<https://www.youtube.com/watch?v=dmtcug2ptg4>
- Gesture Sensor Intelligent Control Programming Dancing Walking:
<https://www.youtube.com/watch?v=2QGGF4cXZ2I>

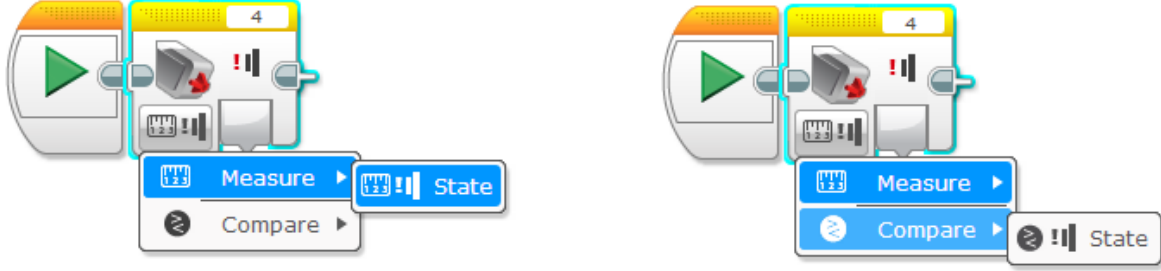
Eğitim boyunca EV3 robot setinde bulunan dört temel sensör sırasıyla anlatılacaktır: touch (dokunma), ultrasonik (mesafe), jiroskop (açı) ve colour (renk). Sensörler öğrencilere gösterilerek kısaca bilgi verilir.

Bu hafta dokunma sensörü tanıtılacaktır. Dokunma sensörü, düğmesine ne zaman basıldığını ya da bırakıldığını, düğmeye bir defa mı yoksa birden fazla mı basıldığını algılayabilen basit ama son derece hassas bir araçtır. Öğrenciler “başla” ve “durdur” gibi basit komutlar vererek robotları bu sensör aracılığı ile hareket ettirebilecekleri gibi labirent benzeri karmaşık ortamlarda robotlarını dokunma sensörü ile yönlendirebilirler.



Resim 49. Dokunma Sensörü

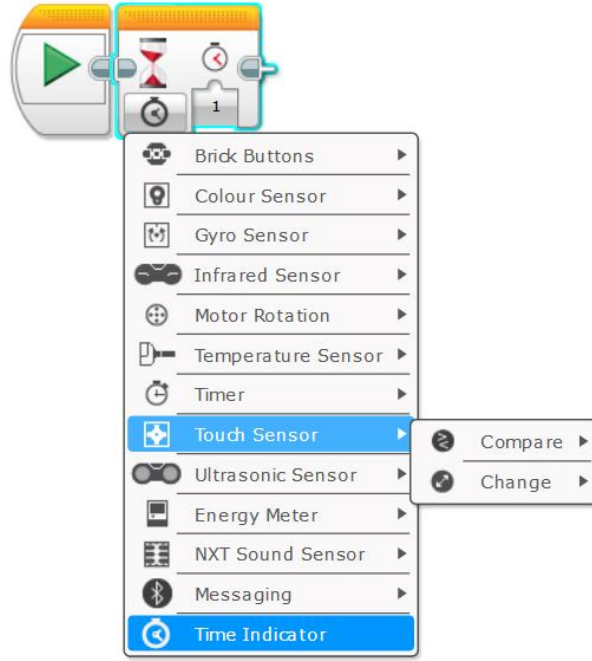
Bu aşamada EV3 yazılımının sensor sekmesinde bulunan dokunma sensörü komutu incelenir. Dokunma sensörü varsayılan olarak 1. porta takılır. Eğer sensör robot üzerinde farklı bir porta bağlanmış ise, kod bloğunun sağ üst köşesinde yer alan port sekmesinden bağlı olduğu port değiştirilir. Dokunma sensöründe temel olarak Measure (ölçüm) ve Compare (karşılaştır) olmak üzere iki menü bulunur.



Resim 50. Dokunma Sensörü Ayarları

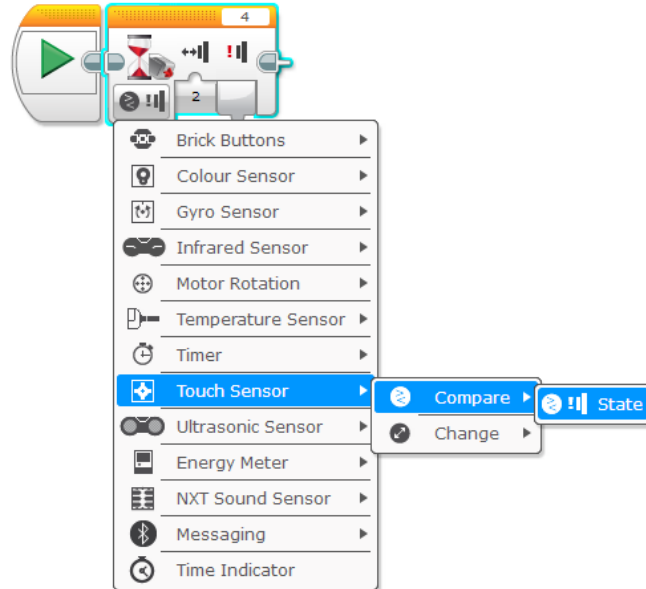
Ölçüm özelliğiyle sensör kontrol edilir. Dokunma sensörünün “bırakıldı” (0), “basıldı” (1) ve “basıldı-bırakıldı” (2) durumları ölçülür ve bu değerler başka bloklara aktarılabilir. Karşılaştır özelliğiyle ise sensörden ölçülen değer, var olan üç durum ile karşılaştırması yapılır ve elde edilen Doğru/Yanlış (True/False) mantık sonucu başka bloklara aktarılabilir.

Sensörlerle ilgili uygulamalara başlamadan önce “akış kontrolü (flow control)” sekmesinden “bekle (wait)” bloğu tanıtılır. Bekle bloğu, EV3 yazılımındaki en basit bloklardan biridir ve en basit akış kontrol yapılarından birini sağlar. Temel işlevi, bir programın yürütülmesini belirli bir koşul sağlanana kadar duraklatmak ve ardından programı devam ettirmektir. Kum saati ikonu ile gösterilen bekle bloğu, sensörler arasında veri akışının sağlanmasında kullanılır. Örneğin bekle bloğu bir sensörün belirli bir değere ulaşması veya bir sensörün değerinin değişmesi için bir süre beklenmesini sağlar.



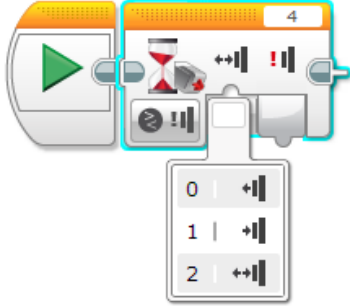
Resim 51. Bekle/Wait Bloğu

Bu aşamada, dokunma sensörüne dokunulunca harekete geçen bir robot programlanır. Bu program için Akış Kontrolü (Flow Control) sekmesinden “bekle (wait)” bloğu ve Eylem (Action) sekmesinden “direksiyon hareketi (move steering)” bloğu kullanılır. İlk önce bekle bloğu program alanına sürüklenip bırakılır. Daha sonra resimde görüldüğü gibi “Touch Sensor>Compare> State” modu seçilir.



Resim 52. Bekle/Wait Bloğu Dokunma Sensörü State Modu

Dokunma sensörünün üç durumu olduğu burada gösterilir.



Blok üzerinde yer alan butona tıklandığında üç farklı fonksiyon bulunduğu görülür:
 0 – Bırakıldığında
 1 – Basıldığında
 2 – Basılıp bırakıldığında

Resim 53. Dokunma Sensörü Ayarları

Bekle bloğunun dokunma sensörü butonuna tıklanır ve durumu 2 yani “basılıp bırakıldığında” olacak şekilde ayarlanır. Daha sonra “direksiyon hareketi (move steering)” bloğu sürüklenip program alanına eklenir ve “motor 5 sn boyunca çalışsın ve sonra da dursun” olacak şekilde ayarlanır. Öğrencilere robotun dokunma sensörüne dokundukları anda harekete geçmeyeceği, ellerini çektikten sonra harekete geçeceği vurgulanır.



Resim 54. Örnek Program

1.2. Uygula: Engele Çarpınca Geri Döner Robot

Öğrencilerden engele çarpınca geri döner bir robot programlamaları istenir. Bunun için aşağıdaki adımlar takip edilir:

- Öğrencilerden robotun bir engele çarpıncaya kadar düz gitmesi istenir.
- Robotun engele çarpınca biraz geri gitmesinin programa eklenmesi istenir.
- Robotun kendi etrafında dönmesinin programa eklenmesi istenir.



Resim 55. Örnek Program

Bu süreçte rehber öğretmen sınıfta dolaşarak öğrencilere yardımcı olur. Aşağıdaki şekilde gösterilen programı oluşturmaları için öğrencileri yönlendirir. Fakat öğrencilerden gelen farklı ve mantıklı fikirleri de değerlendirir ve eğer uygunsa öğrencileri kendi fikirleri doğrultusunda programı oluşturmaları için cesaretlendirir.

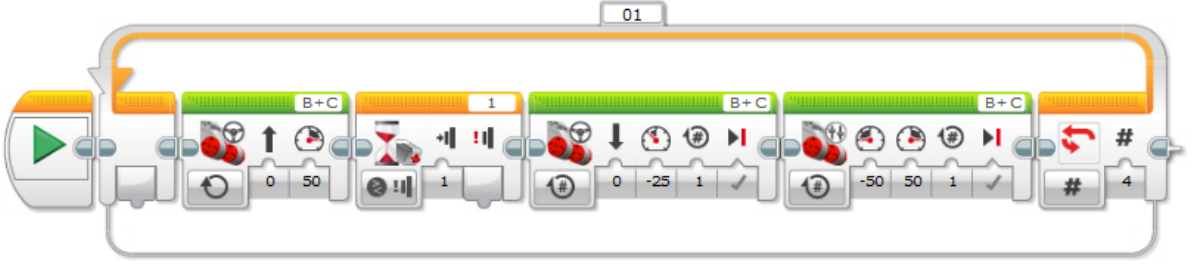
Not

Öğrencilerden robot tasarımlarına dokunma sensörünü de eklemeleri istenir. Ayrıca geçen hafta birleştirilmiş olan kolun sökülmemesi gerekir. Bunun için gerekli yönergeye aşağıdaki yollarla ulaşılabilir:

- i) <https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-touch-sensor-driving-base-4b82858ad3054e725caf23fffde42194.pdf>
- ii) EV3 yazılımı > Lobby > Building Instructions > Building Ideas > Touch Sensor Driving Base
- iii) Robot setiyle birlikte gelen kitapçığa bakılabilir.

1.3. Uygula: Engellere Çarpan ve Her Seferinde Geri Dönme İşlemini Yapan Robot

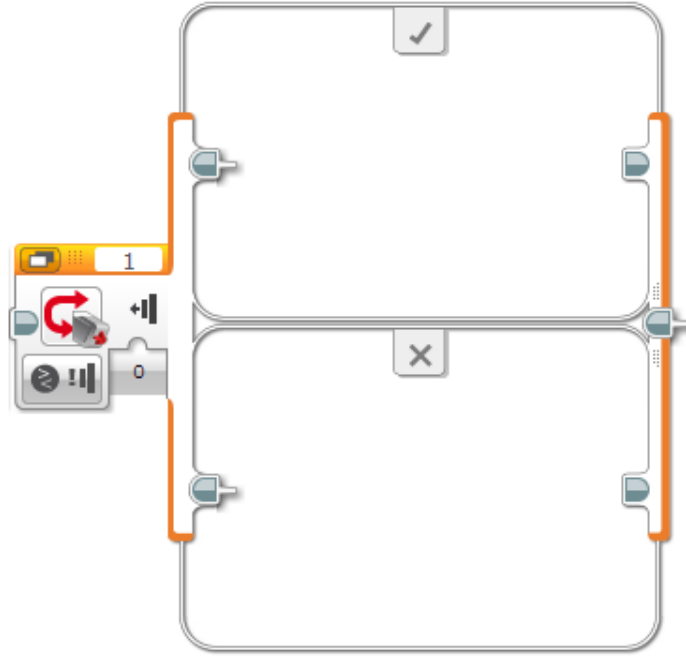
Bu uygulama için öğrencilerden ilk uygulamaya “döngü (loop)” bloğunu eklemeleri istenir. Başka bir ifadeyle öğrencilerden önceki uygulamaya benzer bir program oluşturmaları ve önceki uygulamada bir kez gerçekleştirilen görevin dört defa gerçekleştirilmesini sağlamaları istenir. Bu arada bir işlemin belirli sayıda, sonsuza kadar veya belirli şartlar gerçekleştiğinde devam etmesi istenildiğinde “döngü” kavramının kullanıldığından tekrar bahsedilir. Bu uygulamada robot hareket ederken engele çarparsa dönüş yapar, sonra hareket etmeye devam eder, yine engele çarparsa tekrar dönüş yapar ve bu işlemi dört kez tekrarlar.



Resim 56. Örnek Program

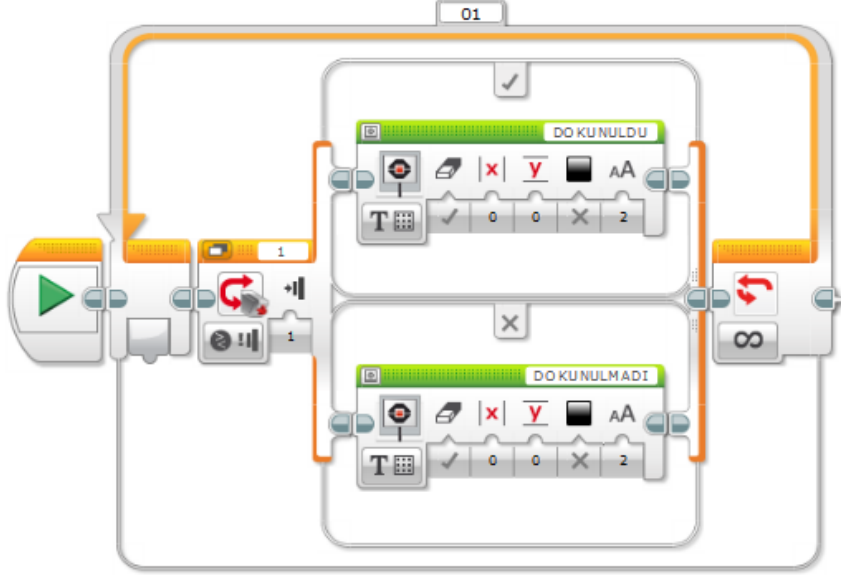
1.4. Gözle: Anahtar (Switch) Bloğu

Öğrencilere, programlama aşamasında belirli işlerin çeşitli koşulların sağlanması durumunda gerçekleşmesinin istenebileceğinden bahsedilir. Örneğin hareket ederken hızlanan bir robotun belirli bir hızdan sonra ses çıkarması, ışık yayması ya da algıladığı renge göre farklı işlemler yapması gibi. EV3 yazılımının “akış kontrolü (flow control)” sekmesindeki “anahtar (switch)” bloğu bu işlemler için kullanılır. Aşağıdaki şekilde görüldüğü gibi “doğru/yanlış” durumuna göre üstteki ya da alttaki işlemler gerçekleştirilir.



Resim 57. Koşul/Switch Bloğu

Switch bloğu kullanılarak robot programlanmaya başlanır. Amaç, dokunma sensörüne basılıp basılmadığını ekranda gösteren bir program oluşturmaktır. Bu program için Eylem (Action) sekmesinden “ekran (display)” bloğu, Akış Kontrolü (Flow Control) sekmesinden “döngü (loop)” ve “anahtar (switch)” blokları kullanılır. İlk önce döngü ve anahtar blokları program alanına sürüklenip bırakılır. Daha sonra koşulun doğru ve yanlış olma durumuna göre ekran “display” blokları “DOKUNULDU” ve “DOKUNULMADI” olarak ayarlanır. Öğrencilerin robotun dokunma sensörüne basılıp basılmadığını yazdıran durumu görmesi sağlanır.

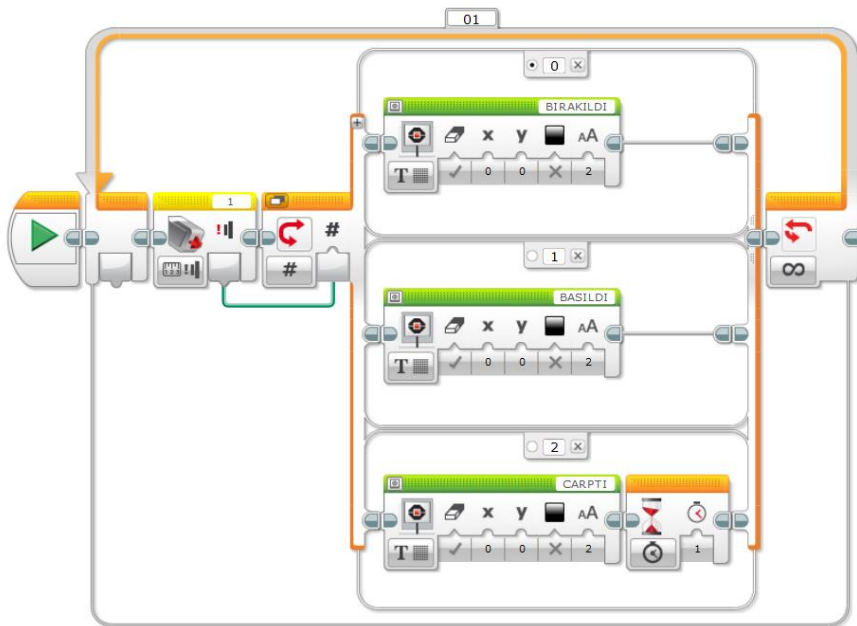


Resim 58. Örnek Program

1.5. Uygula: Koşul Durumunu Göster

Öğrencilere dokunma sensörünün üç durumu tekrar hatırlatılır. Daha sonra ekranda “BASILDI”, “BIRAKILDI” ve “ÇARPTI” ifadelerini gösteren bir program oluşturmaları istenir.

Programı öğrencilerin yapmasını sağlayınız. Öğrencilerin zorlandıkları durumlarda onlara destek olunuz. Öğrenciler programı tamamladıktan sonra, “anahtar (switch)” bloğunun “Sayısal (Numeric)” modunun ve sarı sekmede yer alan dokunma sensörü bloğunun neden kullanıldığını öğrenciler ile tartışınız.



Resim 59. Örnek Program

2. TASARLA

2.1. Oyuncak Köpeğe Çarpınca Havlayan Robot

Bu etkinlikte öğrenciler yaklaşık 50 cm uzaklıktaki oyuncak bir nesneye (örneğin köpeğe) çarpınca havlama sesi çıkarıp ekranında farklı bir görsel göstererek duran bir robot tasarlar. Öğrenciler grup olarak tartışır. Gerekli noktada rehber öğretmen onlara yardımcı olabilir. Fakat öğrencilere tam bir çözüm verilmemelidir. Gruplar çözümü kendileri üretmelidir. Tasarlama için öğrencilerin aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmesi gerekir.

Tanımlama: Öğrenciler öncelikle takip işlemi için neler gerektiğini belirlemeli ve gerekli işlemleri maddeler hâlinde yazmalıdır.

Örneğin;

- Robot sürekli hareket edecek,
- Karşısına bir nesne (örneğin oyuncak köpek) çıkıp çıkmadığı kontrol edecek,
- Nesneye çarpınca havlama sesi çıkaracak,
- Ekranda farklı bir görsel gösterecek,
- Herhangi bir nesne çarpınca duracak.

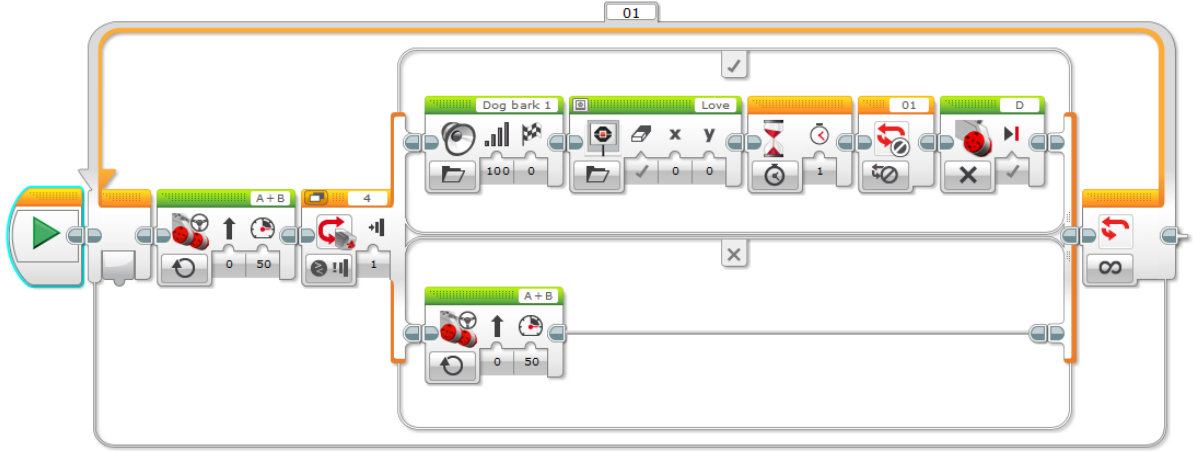
Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi beklenir. Örnek olarak öğrenciler aşağıdaki maddelere benzer fikirler üretebilir:

- Robotun sürekli ilerlemesi ve önünde herhangi bir nesnenin olup olmadığını kontrol etmesi için dokunma sensörü ve “döngü (loop)” bloğu kullanılabilir.
- Robotun nesneye çarpıp çarpmadığını kontrol etmek için “anahtar (switch)” bloğu kullanılabilir.
- Nesneye çarptığında, “ekran (display)” bloğu ve “ses (sound)” bloğu kullanılarak, robotun ekran görüntüsünün değişmesi ve havlama sesi çıkarması sağlanabilir.
- Nesneye çarpmadığı sürece robotun sürekli ilerlemesi sağlanabilir.

Öğrenciler isterse farklı nesnelere kullanarak robotun birden fazla ses çıkarmasını sağlayabilirler. Her grubun çözümü farklı olabilir, önemli olan bu fikirlerin gruplar tarafından ortaya konulması ve ortaya konulan fikirlerin problemin çözümünü sağlayabilmesidir.

3. ÜRET

Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenilen görevi yerine getirir. Program rehber öğretmenlere verilecektir, örnek çözümün bir kısmı aşağıda görülebilir:



Resim 60. Örnek Program

4. DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Aşağıdaki sorular üzerinden tartışma ortamı yaratılır.

- Bugün yapılan etkinlikte sizce neden motorlar için “on” durumu kullanıldı?
- Neden “anahtar (switch)” bloğu kullanıldı?
- Dokunma sensörünün günlük hayatta kullanım örnekleri nelerdir? (Örneğin arabanın kapısı açık kaldığında uyarı amacıyla ses vermesi)

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilebilir.

5. İLAVE ETKİNLİK

5.1. Çift Sensör Kullanımı

Bu etkinlikte öğrencilere iki dokunma sensörünü tek bir robotta kullanacakları söylenir. Amaç, hareket hâlindeki robotun dokunma sensörüne tıklanıldığında robotun dokunulan sensörün yönüne doğru hareket etmesini sağlamaktır.

Tanımlama: İlave etkinliklerde de tanımlama ve fikir yürütme adımlarının gerçekleştirilmesi öğrencilere algoritma yazma alışkanlığı kazandıracaktır.

Örneğin;

- Robot sürekli hareket edecek,
- Sensöre dokunulup dokunulmadığını kontrol edecek.
- Sensöre dokunulursa dokunulan sensör yönünde 90 derece dönecek.

Fikir üretme: Aşağıdaki maddelere benzer fikirler üretebilir:

- Robotun sürekli ilerlemesi için “döngü (loop)” bloğu kullanılabilir.

- Sensöre dokunulup dokunulmadığını kontrol etmek için “anahtar (switch)” bloğu kullanılabilir.
- Sensöre dokunulursa dokunulan yönde robotun 90 derece dönmesi sağlanabilir.

5. Hafta: Mesafe Sensörü

Ön bilgi:

- Öğrenciler, temel düzeyde robot kavramını bilir.
- Öğrenciler, robot setiyle farklı robot tasarımları yapmıştır.
- Öğrenciler, robot setini programlamak için grafik ara yüzünü kullanmıştır.
- Öğrenciler, robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler, tuğlanın çeşitli sesleri çıkarması için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler, tuğla ekranının görüntüsünü düzenlemek için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler dokunma sensörünü farklı amaçlar için programlama adımlarını oluşturmuştur.

Haftanın Kazanımları:

- Öğrenciler, mesafe sensörünün çalışma mantığını ve kullanma yöntemlerini ifade eder.
- Öğrenciler, mesafe sensörünü farklı amaçlar için programlama adımlarını oluşturabilir.
- Öğrenciler, robotun farklı mesafelerdeki nesnelere göre davranması (hareket etmesi, ses çıkarması, hızını ayarlaması, tuğla ekranında farklı simgeler göstermesi) için gerekli programlama adımlarını oluşturabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin mesafe (ultrasonic) sensörünün çeşitli kullanım şekillerini kavrayarak sensörü farklı görevler için programlarken gerekli düzenlemeleri yapabilmesini sağlamaktır.

Kullanılacak Malzemeler:

Robot seti, bilgisayar, mesafe sensörü, mat (çalışma alanı).

Haftanın İşlenişi:

Gözle: Mesafe sensörünü kullanarak robota belirli mesafeye göre işlem yaptırma, belirlenen mesafe kadar hareket ettirme, sensörü kullanarak teker yarıçapını ölçme

Uygula: Her bir komutun çeşitli bileşenlerini programlayarak uygulama

Tasarla: Robotun istenilen işlemleri yapabilmesi için gerekli bileşenlerini tanımlama ve planlama

Üret: Verilen görevleri programlama

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği

1. GÖZLE ve UYGULA

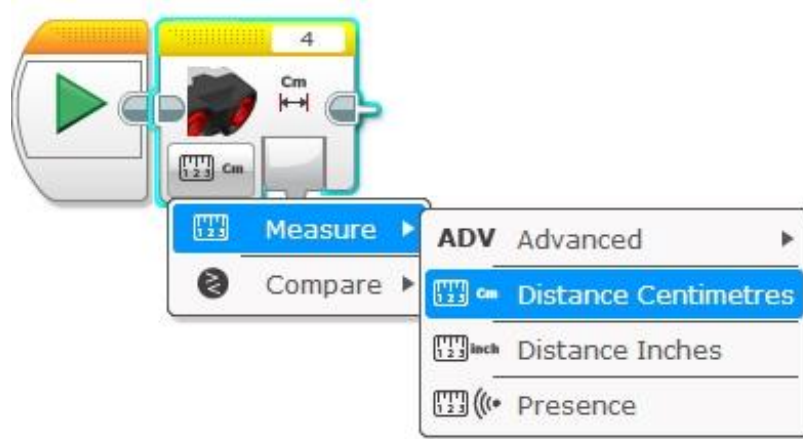
1.1. Gözle: Mesafe Sensörü Nedir, Nasıl Kullanılır?

Mesafe sensörü (ultrasonic sensor) ileriye doğru yüksek frekanslı ses dalgaları gönderir ancak bunlar insanlar tarafından duyulamaz. Karşıda bulunan nesneye çarpan ses dalgaları yansyarak geri döner ve mesafe sensörü tarafından geri alınır. Ses dalgasını gönderme ve alma arasında geçen süre kullanılarak mesafe sensörünün karşıdaki nesneye uzaklığı hesaplanır. Mesafe sensörü ile nesne arasındaki mesafenin başlangıç noktası sensör üzerindeki “gözler” değildir, mesafe resimde gösterilen noktadan başlanarak bulunur. Mesafe sensörü en fazla 255 cm uzaklıktaki cisimleri algılayabilir, daha uzaktaki cisimleri algılayamaz. Ayrıca, mesafe sensörüyle santimetre cinsinden veya inç cinsinden uzaklık işlemleri yapılabilir.



Resim 61. Mesafe Sensörü

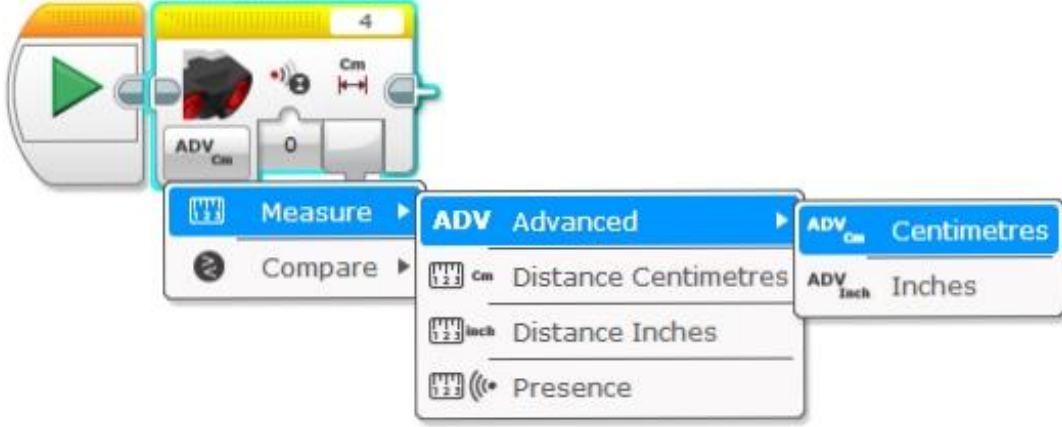
EV3 yazılımında sensör sekmesinde “mesafe sensörü (ultrasonic sensor)” komutu bulunur. Aşağıda resimde görüldüğü üzere mesafe sensörü dördüncü porta takılır. Mesafe sensöründe temel olarak “Ölçüm (Measure)” ve “Karşılaştır (Compare)” olmak üzere iki menü bulunur.



Resim 62. Mesafe Sensörü Bloğu

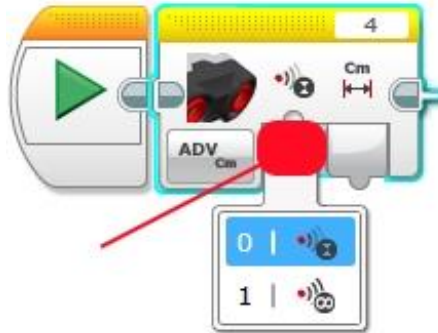
Ölçüm menüsü kullanılarak karşıda bulunan cisme santimetre cinsinden uzaklık (distance centimeters) ve inç cinsinden uzaklık (distance inches) belirlenebilir, ayrıca karşıda başka bir mesafe sensörünün olup olmadığı (presence) sorgulanabilir. Santimetre veya inç cinsinden hesaplamada uzaklık değeri sayısal olarak verilirken

“presence” komutu ile doğru veya yanlış olmak üzere “boolean” bir değer üretilir. Karşıda başka bir mesafe sensörü varsa “doğru”, yoksa “yanlış” değeri üretilir. “Gelişmiş (Advanced)” menüsü de aşağıda resimde görüldüğü gibi, santimetre ve inç cinsinden uzaklık ölçmek için kullanılır.



Resim 63. Mesafe Sensörü Ölçüm Seçenekleri

Gelişmiş (Advanced) menüsünden santimetre komutu verildiğinde aşağıdaki resimde görünen menü ortaya çıkar. Fare ile aşağıda kırmızı ile işaretlenmiş yere tıklandığında iki seçenek çıkar. Bunlar 0 (ping) ve 1'dir (continuous). 0 seçeneği ile karşıya sadece bir tane ses dalgası gönderilirken 1 seçeneği ile sürekli ses dalgası gönderilir. Bu seçenekler bazı durumlarda faydalı olabilir. Örneğin ortamda birden fazla mesafe sensörü varsa her iki sensörden gelen sürekli ses dalgaları yanlış hesaplamalara sebep olabilir. Bunu engellemek için sensörlerin sırayla birer tane dalga göndermesi sağlanabilir.



Resim 64. Advance Menüsü Seçenekleri

1.2. Gözle: Belirli Bir Mesafeye Kadar İlerleme

Bu program için Eylem (Action) sekmesinde bulunan Direksiyon Hareketi (Move Steering) ve Akış Kontrolü (Flow Control) menüsünde bulunan “Bekle (Wait)” komutları kullanılır. Direksiyon hareketi komutu robotun iki motorunun birlikte hareket ettirilmesi için kullanılır. “Bekle” komutu ise belirli bir olay gerçekleşinceye kadar belirli komutların

çalıştırılmasını sağlar. Öncelikle robota mesafe sensörü takılır. Programda aşağıdaki resimde görülen komutlar oluşturulur.



Resim 65. Örnek Program

Not

Mesafe sensörlü temel tasarıma (ultrasonic sensor driving base) ulaşmak için gerekli yönergeye aşağıdaki yollarla ulaşılabilir:

- i) <https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-ultrasonic-sensor-driving-base-61ffdfa461aee2470b8ddb8db16e2070.pdf>
- ii) Ev3 yazılımı > Lobby > Building Instructions > Building Ideas > Ultrasonic Sensor - Driving Base
- iii) Robot setiyle birlikte gelen kitapçık kullanılabilir.

Görüldüğü üzere, oluşturulan kod ile B ve C portlarına takılan büyük motor'ların 3 numaralı porta takılan mesafe sensöründen gelen uzaklık değeri 5 cm'den küçük oluncaya kadar çalıştırılıp mesafe 5 cm'den küçük olduktan sonra durması sağlanır. Motorlar 50 hızıyla hareket edip doğruca karşıya ilerler.

Bu adımda programın nasıl yazıldığı ekrana yansıtılarak öğrencilere gösterilir ve programın çalışma mantığı öğrencilere açıklanır.

Rehber öğretmen burada aşağıdaki adımların öğrenciler tarafından yapılması ister

- (i) Yukarıdaki kod EV3 yazılımında yazılır ve robot çalıştırılır.
- (ii) Kodun çalışması bitince nesne ile mesafe sensörü arasındaki mesafe ölçülür.
- (iii) Öğrencilerden buldukları değer ve programda yazılı olan mesafe değerini (ilk örnek için 5 cm) bir kâğıda yazmaları istenir.
- (iv) Öğrencilerden i, ii ve iii adımları 7 cm, 10 cm ve 15 cm için tekrarlaması istenir.
- (v) Öğrencilere buldukları verilerin ne anlama geldiği sorulur.
- (vi) Öğrencilere robotun tam olarak verilen değerde duramayacağı ve hata payı bulunduğu için çıkan mesafe değerinin verilen değerlerden farklı çıktığı açıklanır.

1.2. Uygula: İstenilen Mesafe Kadar Geri Gitme

(i) Öğrencilerden mesafe sensörünü kullanarak, robotlarının mesafe sensörünün önünde bulunan engelden 20 cm geriye gittikten sonra durmasını sağlayacak programı oluşturmaları istenir.

(ii) Rehber öğretmen sınıfta dolaşarak öğrencilere yardımcı olur. Öğrenciler aşağıdaki resimde gösterilen programı oluşturmaları için yönlendirir. Fakat öğrencilerden gelen farklı ve mantıklı fikirleri de değerlendirir ve eğer uygunsa kendi fikirlerindeki programı oluşturmak için öğrencileri cesaretlendirir.

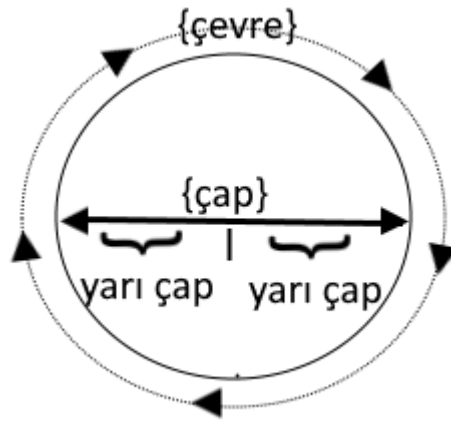


Resim 66. Örnek Program

(iii) Öğrencilerden programı çalıştırdıktan sonra çıkan mesafeyi ölçmeleri istenir. Bu mesafeyi 20 cm'ye olabildiğince yaklaştırmaları için programda değişiklik yapmaları istenir.

1.3. Gözle: Tekerin Yarıçapını Hesaplama

Öğrencilere dairenin yarıçapı ile çevresi arasındaki ilişki sorulur. Öğrenciler " $\text{çevre} = 2\pi \cdot r$ " formülünü biliyorsa bir sonraki adıma geçilir. Eğer bilmiyorlarsa formülün ne anlama geldiği görsel kullanılarak açıklanır. Öğrencilere ayrıca çemberin çevresinin çapına oranının sabit olduğu ve bunun $22/7$ sayısına yakın olduğu ve bu sonucun " π (pi)" sayısı olarak adlandırıldığı söylenir. Bu seviyede öğrencilerin " π " sayısı ve çemberin çevresi ile ilgili olarak daha detaylı bilgiye ihtiyacı olduğu düşünülmemektedir.



Resim 67. Çevre, Çap ve Yarı Çap

Öğrencilere tekerin aşağıdaki algoritmanın kullanılabileceği anlatılır.

yarıçapının bulunması için

(a) Karşıda bulunan engel ile aradaki mesafe ölçülür (ilk ölçüm),

- (b) Tekerler bir tur ileri döndürülür,
- (c) Mesafe tekrar ölçülür (ikinci ölçüm),
- (d) Mesafe farkı hesaplanır (fark = ilk ölçüm – ikinci ölçüm),
- (e) Fark 2π 'ye (π 3.141 alınacaktır) bölünür ($2\pi = 2 * 3.141 = 6.282$) ve yaklaşık yarıçap değeri bulunur (yarıçap = fark / 6.282).

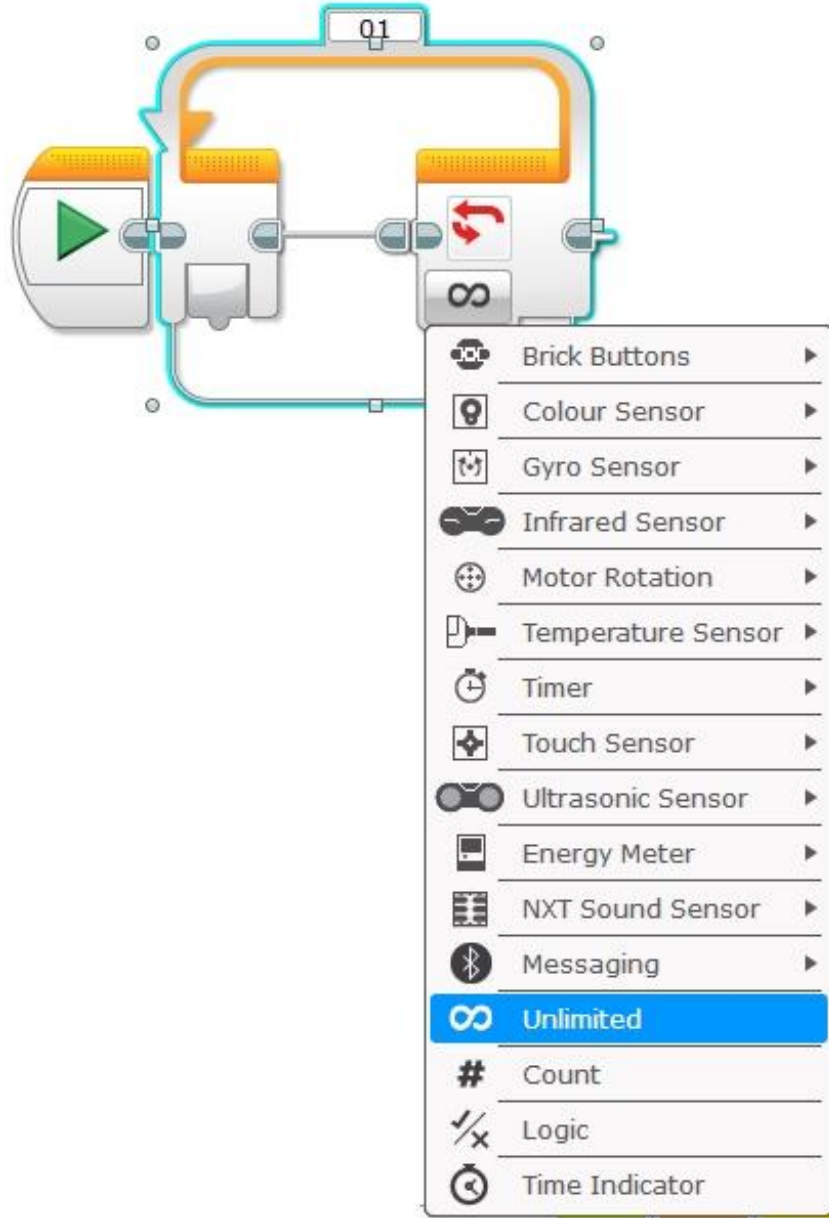
1.4. Uygula: Tekerin Yarıçapını Hesaplama

Öğrencilerden tekerin yarıçapını oluşturacakları program ile hesaplamaları istenir (**Not:** Yarıçap yaklaşık 28 milimetre olarak bulunmalıdır).

Öğrenciler yarıçapı kendileri bulamazlarsa rehber öğretmen tüm sınıf karşısında program yardımıyla yarıçapı hesaplar.

1.5. Uygula: Engele Yaklaştıkça Yavaşlayan Robot

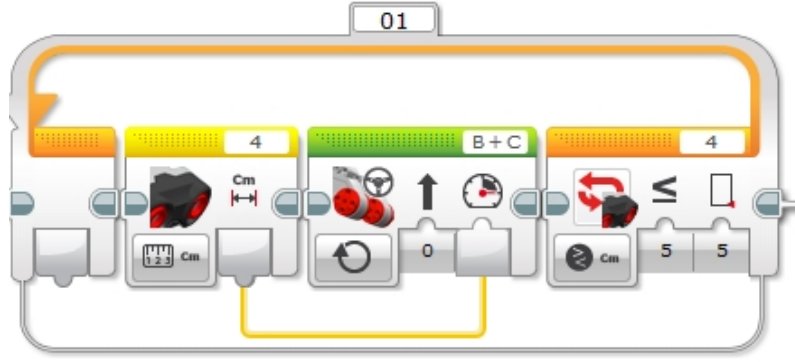
(i) Bu etkinlikte karşıda bulunan engele yaklaştıkça yavaşlayan ve engele 5 cm kaldığında duran bir robot programı yapılacaktır. Bu programın yazılması için turuncu Akış Kontrolü (Flow Control) sekmedeki “döngü (loop)” komutunun bilinmesi gerekir. Rehber öğretmen öğrencilere aşağıda resmi olan “döngü” komutunu gösterir ve seçenekleri hızlıca anlatır. Ses, metin ve grafik konularının işlendiği üçüncü hafta gösterilen “döngü komutu” burada da tekrar edilir.



Resim 68. Döngü Bloğu Seçenekleri

(i) Öğrencilere mesafe sensörü ile ölçülen uzaklık değerinin, robotun hareketini sağlayan motorların güç değeri olarak aktarıldığında, mesafe azaldıkça motorların gücünün de azalacağı ve böylece robotun engele yaklaştıkça yavaşlayacağı detaylıca anlatılır,

(ii) Aşağıdaki resimde görülen program oluşturularak çalışma mantığı öğrencilere anlatılır.



Resim 69. Örnek Program

2. TASARLA VE ÜRET

2.1. Birinci Görev: Öndeki Aracı Takip Eden Robot

Bu etkinlikte robotun önünde bulunan bir arabayı (araba olmak zorunda değil herhangi bir cismi) takip etmesi sağlanacaktır.

Öğrencilerden öndeki arabayı takip eden, yani öndeki araba ilerledikçe ilerleyen bir robotun program kodunun nasıl yazılacağı üzerinde düşünmeleri istenir. Öğrenciler grup olarak tartışılır. Gerekliğinde rehber öğretmen onlara yardımcı olabilir. Yine öğrencilere tam bir çözüm verilmemelidir. Gruplara çözümü kendi kendilerine üretmesi için zaman verilmelidir. Ders programı boyunca, tasarlama sürecinde olduğu gibi, öğrencilerin aşağıda örnek olarak da verilen iki adıma benzer bir süreci gerçekleştirmeleri gerekir.

Tanımlama: Öğrencilerin öncelikle istenilen takip işleminin neler gerektirdiğini belirlemesi ve ortaya koyması gerekir. Öğrenciler gerekli işlemleri maddeler hâlinde yazar. Örneğin;

- Robot öndeki araç ile arasındaki mesafeyi sürekli kontrol edecek.
- Öndeki araç hareket ederse hareket edecek.
- Öndeki araç hızlı hareket ediyorsa robot hızlanacak, yavaş hareket ediyorsa yavaşlayacak.
- Öndeki araç durunca duracak.

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdaki maddelere benzer fikirler üretebilir.

- Öndeki araçla arasındaki mesafeyi sürekli kontrol etmek için mesafe sensörü ve döngü kullanılabilir.
- Öndeki araçla arasındaki mesafe 20 cm'den büyükse motorların gücü 10 olarak ayarlanacak.
- Öndeki araçla arasındaki mesafe 40 cm'den büyükse motorların gücü 20 olarak ayarlanacak.
- Öndeki araçla arasındaki mesafe 60 cm'den büyükse motorların gücü 50 olarak ayarlanacak.

- Öndeki araçla arasındaki mesafe 80 cm'den büyükse motorların gücü 100 olarak ayarlanacak.
- Öndeki araçla arasındaki mesafe 20 cm'den küçükse motorların gücü 10 olarak ayarlanacak.
- Öndeki araç 10 cm'den yakınsa robot duracak.

Temel olarak bu adımlarla robottan istenilen işlemler yapılabilir. Ama öğrenci isterse robotun öndeki araca daha yakın gitmesini veya öndeki aracın hızına daha fazla ayak uydurmasını sağlayabilir. Mesafeye bağlı olarak robotun yavaşlaması da sağlanabilir. Her grubun çözümü farklı olabilir, önemli olan bu fikirlerin gruplar tarafından ortaya konulması ve ortaya konulan fikirlerin problemin çözümünü sağlayabilmesidir. Rehber öğretmen öğrencilerden gelebilecek olan farklı görüş ve çözüm önerilerine açık olmalı ve onları bu konuda teşvik etmelidir. Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenilen görevi yerine getirmelidir.

2.2 İkinci Görev-Takımlar Yarışıyor

Bu bölümde iki yarışma yapılır. Öğrencilerin yarışmalara takım olarak katılması gerekir. Yarışmalardaki amaç robotun ileride bulunan bir engele kadar ilerlemesi ve geri gelerek başlangıç noktasına vardığında durmasıdır. Başlangıç noktasından 1 metre uzağa engel konulur. Robotların engele 10 cm mesafe kala geri dönmesi ve başladıkları yere varması gerekir. Robot engele 8 cm'den fazla yaklaşırsa veya 12 cm'den daha uzak bir mesafedeyken geri gitmeye başlarsa yarışmacılar elenir. Ayrıca robot geri geri giderken ilk başlangıç noktasını 2 cm'den fazla geçer ya da başlangıç noktasına 2 cm'den daha fazla mesafe varken durursa yarışmacılar yine elenir.

Yarışma 1: Bu yarışmada amaç başlangıç çizgisine en yakın durmaktır. Programı yazmaya başlamadan önce grupların tasarlama adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmesi gerekir. Bu yarışma için verilen süre rehber öğretmen tarafından belirlenir. Yukarıda belirtilen kurallarla birlikte bu süreyi geçiren yarışmacılar elenmiş sayılır. Bütün grupların robotları işlemlerini bitirdiğinde başlangıç noktasına olan uzaklıkları ve işlemi tamamlama süreleri kayıt altına alınmalıdır. Başlangıç çizgisine en yakın duran robot yarışmayı kazanır. Eşitlik durumunda ise hızlı gelen robot birinci sayılır.

Yarışma 2: Bu yarışmadaki amaç görevi en kısa sürede tamamlamaktır. Programı yazmaya başlamadan önce grupların tasarlama adımı için yine yukarıda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmesi gerekir. Yarışma kurallarına uyan robotlardan görevi en hızlı tamamlayan birinci olur. Eşitlik durumunda başlangıç noktasına daha yakın olan robot birincili sayılır.

3. DEĞERLENDİR

Bu aşamada hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşüncelerini sağlamaktır. Bu sayede öğrenciler; problem çözme yetenekleri, dersin konusu ve kendileri ile ilgili gözlemler yaparak öğrendikleri yeni konuları ve kendilerini değerlendirmekle beraber sonraki çalışmalarını planlamak için de fırsat elde edeceklerdir. Öğrencilerden şu soruları yanıtlamaları istenir:

- Size bugün tasarla ve üret adımında verilen problemler nelerdir? (Problemi kendi cümleleri ile ifade etme).
- En çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa, rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
 - Öndeki aracı takip eden programda, robotun hareket etmeye başlamasını nasıl sağladınız?
 - Öndeki aracı takip eden programda, öndeki araç durunca robotun durmasını nasıl sağladınız?
 - İkinci görevde, belirlenen sürede robotu engele yaklaştırıp sonra başladığı noktaya geri getirmek için ne gibi çözümler denediniz?
- Problemi çözerken ne gibi sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
- Kullandığınız yöntemler bu sıkıntıları gidermede başarılı oldu mu?
- Grup arkadaşınızla anlaşmazlığa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne/neler öğrendiniz?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilebilir.

4. İLAVE AKTİVİTE

4.1. Park Sensörü

Öğrencilerden robotu taşıtlardaki park sensörüne benzer şekilde programlamaları istenir. Robotun belirli bir mesafeye gelince uyarı vermesi, mesafe azaldıkça sesi yükseltmesi ve sesin tekrar aralığının artması, tuğla ekranında mesafe ve uyarı ile paralel görseller yansıtması istenir.

Tasarla ve Üret: Öğrenciler program adımlarını detaylı olarak planlamalı (mesafeleri belirlemeli, belirlenen mesafelerdeki işlemleri tanımlamalı, kullanılacak sesleri ve görselleri seçmeli, mesafeler değiştiğinde robotun nasıl davranacağına karar vermeli) ve bu öğelerin programla nasıl yapılabileceğini tasarlamalıdır. Bu ilave etkinlikte de öğrenciler aktif rol üstlenir. Rehber öğretmen sadece yönlendirir ve öğrencilere takıldıkları noktalarda destek olur. Üret aşamasında, öğrencilerden bir önceki adımda tasarladıkları robot planını kullanarak probleme “algoritmik” –daha yapılandırılmış- bir çözüm önerisi geliştirmeleri istenir. Öğrenciler bilgisayar ve robot başında çalışarak gerekli yazılım çözümleri geliştirmelidir. Burada öğrencilerin en çok zorlanacağı konulardan birisi, ses, hareket, görüntü ve mesafe sensörünün senkronizasyonunu sağlanmaktadır.

6. Hafta: Açık Sensör

Ön bilgi:

- Öğrenciler robot setiyle farklı robotik tasarımlar yapmıştır.
- Öğrenciler robot setini programlamak için grafik arayüzünü kullanmıştır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler akıllı tuğlanın çeşitli sesleri çıkarması ve ekranının görüntüsünü düzenlemek için gerekli programlama adımlarını oluşturabilir.
- Öğrenciler robotu programlarken “döngü (loop)” ve “anahtar (switch) mantığını kurup uygulayabilir.

Haftanın Kazanımları:

- Öğrenciler, açı (gyro veya jiroskop) sensörünün çalışma mantığını ve kullanma yöntemlerini bilir.
- Öğrenciler, robotun istenilen açılarda dönmesini sağlayan programları açı sensörünü kullanarak oluşturabilir.
- Öğrenciler, açı sensörünü farklı amaçlar için programlama adımlarını oluşturabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin açı sensörünü kullanarak robotun dönme hareketini istenilen açıda gerçekleştirebilmesini temin etmek ve açı sensörünü kullanırken karşılaştıkları problemlerin sebeplerini ve çözümlerini tartışabilmelerini sağlamaktır.

Kullanılacak Malzemeler:

Robot seti, açı sensörü, bilgisayar, mat (çalışma alanı).

Haftanın İşlenişi:

Gözle: Robotu açı sensörü ile yönlendirerek hareket ettirme ve açı sensörünü kalibre etme.

Uygula: Açık sensörünü kullanarak farklı uygulamalar geliştirme.

Tasarla: Açık sensörünü kullanarak robotun düz gitmesini sağlayacak tasarım için gerekli bileşenleri tanımlama ve planlama.

Üret: Verilen görevlere göre robotu programlama.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği.

1. GÖZLE VE UYGULA

1.1. Gözle: Açık Sensör

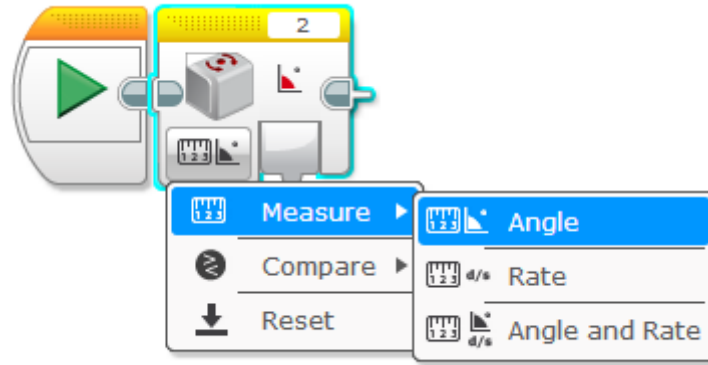
Gyro veya jiroskop olarak da adlandırılan açı sensörü ile robotun hareketi, hareket yönü veya eğimi algılanabilir. Öğrenciler bu sensörle açıları ölçerek navigasyon sistemlerinden oyun kumandalarına kadar çeşitli gerçek dünya araçlarında kullanılan teknolojileri keşfedebilir. Açık sensörünün temel özellikleri şöyle sıralanabilir:

- Üzerindeki ok yönündeki dönüşleri algılar,
- +/-3 derecelik bir hassasiyetle açıları ölçer,
- Saniyede maksimum 440 derecelik açı değişimini algılar,
- Yenileme hızı 1kHz'dir.



Resim 70. Açık Sensör

Bu aşamada EV3 yazılımında sensör sekmesinde bulunan açı sensörü bloğu incelenir. Aşağıdaki resimde görüldüğü üzere açı sensörü varsayılan olarak ikinci porta takılır. Açık sensöründe temel olarak Ölçüm (Measure), Karşılaştır (Compare) ve Sıfırla (Reset) olmak üzere üç menü bulunur.

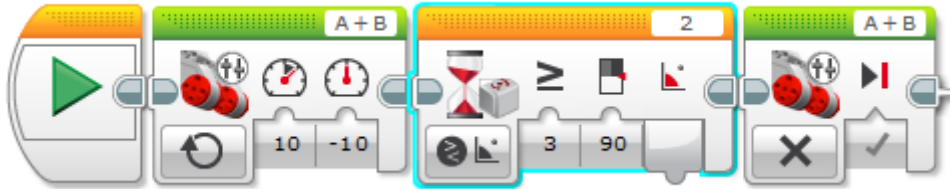


Resim 71. Açık Sensör Ayarları

Ölçüm özelliğiyle sensörün durumu kontrol edilebilir. Açık sensör sıfırlandığı andan itibaren robotun saat ibresi veya saat ibresinin tersi yönde yaptığı dönme miktarını "açık derecesi (angle)" cinsinden ölçer. Ayrıca saniyedeki "açık değişim oranını (rate)" hesaplayabilir. Bu iki ölçüm değeri istenilirse birlikte de alınabilir (Angle and Rate). Karşılaştır (Compare) özelliği ile ölçülen açı ve oran değerleri önceden belirlenmiş bir değeri ile karşılaştırılır ve elde edilen mantık değeri Doğru/Yanlış (True/False) olarak

başka bloklara aktarılabilir. Ayrıca açık sensörünün değerini sıfırlamak gerekirse sıfırla (reset) modu seçilir.

Bu aşamada açık sensörünü kullanarak robotu kodlama işlemine geçilir. Amaç, olduğu yerde 90 derece dönen bir robot yapmaktır. Bu program için Eylem (Action) sekmesinden “palet hareketi (move tank)”, Akış Kontrolü (Flow Control) sekmesinden de “Bekle (Wait)” blokları kullanılır. İlk önce palet hareketi (move tank) bloğu program alanına sürüklenip bırakılır. Motor, açık sensörden gelen değerlere göre hareket edeceği için özelliği “on” olarak değiştirilir. Sonra bekle (wait) bloğu program alanına sürüklenip bırakılır. Ardından Gyro Sensor>Compare> Angle modu seçilir. Son olarak bir tane daha palet hareketi (move tank) bloğu programlama alanına sürüklenip bırakılır ve özelliği “off” olacak şekilde değiştirilerek, motorlar durdurulur. Bu aşamadan sonra uygulamaya geçilir.



Resim 72. Örnek Program

Not

Öğrencilerden robot tasarımlarına açık sensörünü de eklemeleri istenir. Bunun için gerekli yönergeye aşağıdaki yollarla da ulaşılabilir:

- i) <https://le-www-live-s.lego.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-gyro-sensor-driving-base-a521f8ebe355c281c006418395309e15.pdf>
- ii) EV3 yazılımı > Lobby > Building Instructions > Building Ideas > Gyro Sensor Driving Base
- iii) Robot setiyle birlikte gelen kitapçığa bakılabilir.

1.2. Uygula: 360 Derece Dönen Robot

Bu uygulamanın adımları aşağıdaki gibi olabilir:

- (i) Rehber öğretmen öğrencilerden mat üzerinde yer alan ölçerleri kullanarak robotlarının olduğu yerde 360 derece dönmesini sağlayan programı oluşturmalarını ister.
- (ii) Rehber öğretmen dolaşarak öğrencilere yardımcı olmalıdır. Öğrencileri aşağıdaki resimde gösterilen programı oluşturmak üzere yönlendirir. Fakat öğrencilerden gelen farklı ve mantıklı fikirleri de değerlendirir ve uygunsa öğrencileri kendi fikirlerindeki programı oluşturmak için cesaretlendirir.



Resim 73. Örnek Program

(iii) Öğrencilerden programın sonucunu değerlendirmeleri istenir. Genel olarak robotun 360 dereceden biraz daha fazla döndüğü görülür. Robotun neden tam olarak 360 derece dönmediği tartışılır. Robotun 360 dereceden daha fazla dönmesinin nedenleri şunlar olabilir:

- (f) Açık sensörün hassasiyeti 360 derece için +/-3 derecedir. Yani sensör yeterince hassas değildir ve robotun dönme açısı 357 ile 363 derece arasında değişebilir.
- (g) Sensör tam olarak 360 dereceyi ölçtüğü anda bu verinin aktarılması ve motorlara durma komutunun gönderilmesi zaman alır. Bu nedenle robot 360 dereceden fazla döner.
- (h) Robot kendi etrafında dönerken oluşan dönme kuvveti, motorlar durdurulduktan sonra da robotun biraz daha dönmesine sebep olur.

(iv) Öğrencilerin bu sonuçlara ulaşabilmesi için rehber öğretmen tartışma ortamı oluşturur ve öğrencileri bu sonuçlara ulaştıracak sorularla tartışmayı yönlendirir. Sonrasında robotun tam 360 derece dönebilmesi için ne yapılması gerektiği tartışılır. Örneğin, robotun 360 dereceden daha az bir açı ile dönecek şekilde programlanması bir çözüm olabilir.

1.3. Gözle: Açık Sensör Kalibrasyonu

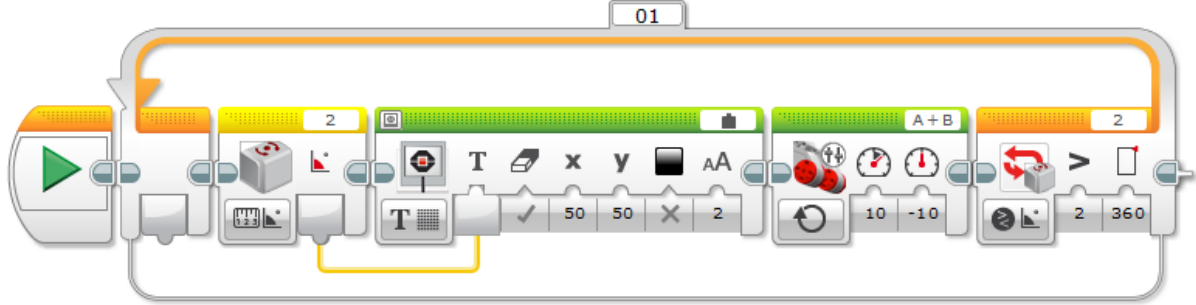
Açık sensöründe bazı kaymalar oluyorsa sensörün kalibre edilmesi gerektiği söylenir. Açık sensörü, robotla olan bağlantısı koparılıp tekrar kablosu takılarak veya port modu değiştirilerek kalibre edilir. Açık sensörünü kalibre etmek için aşağıdaki resimde sunulan program da çalıştırılabilir. Programda açık sensörü önce "oran" (rate) moduna alınır ve bir saniye sonra "derece" moduna alınır. Böylece açık sensör sıfırlanmış (kalibre edilmiş) olur.



Resim 74. Açık Sensör Kalibrasyonu

1.4. Uygula: Açık Sensör Değerini Ekran Yazma

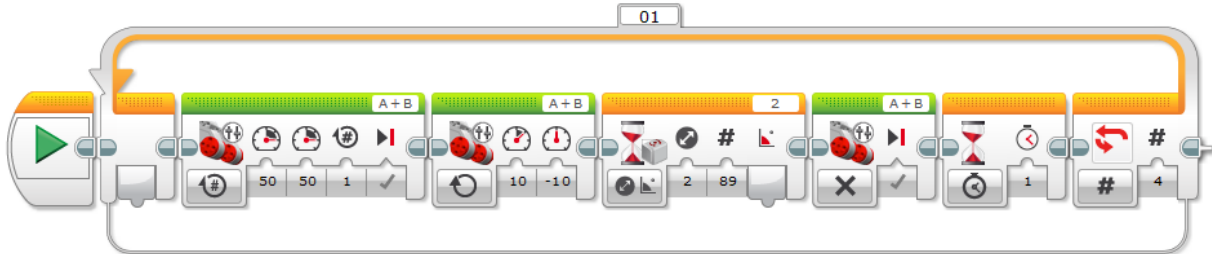
Öğrencilerden robot olduğu yerde 360 derece dönerken açı sensöründen gelen değerlerin tuşla ekranına yazılmasını sağlayan bir program oluşturmaları istenir.



Resim 75. Örnek Program

1.5. Uygula: Açık Sensörünü Kullanarak Kare Şeklinde Hareket Eden Robot

Öğrencilerden robotun kare şeklinde hareket etmesini sağlayacak bir program oluşturmaları istenir. Bu uygulamanın bulunduğu yerde 360 derece dönen robot tasarımına çok benzediği söylenir. Nelerin farklı olması gerektiği tartışılır.



Resim 76. Örnek Program

2. TASARLA

2.1. Açık Sensörünü Kullanarak Robotun Düz Gitmesini Sağlamak

Bu etkinlikte öğrencilerden herhangi bir çizgiyi takip etmeden düz bir şekilde giden robotun dışarıdan yapılacak bir müdahale sonrasında tekrar rotasına dönmesini sağlayan bir robot tasarımları istenir.

Öğrencilerden robotlarını doğrusal ilerleyecek şekilde programlamaları istenir. Fakat doğrusal ilerlerken yönünü sağa veya sola döndürecek bir müdahale sonrasında ilerlediği doğrultuyu tekrar başlangıçtaki doğrultusuna ayarlayabilecek bir robot olması gerektiği belirtilir. Örneğin robot başlangıçta kuzeye doğru doğrusal ilerlerken, rehber öğretmenin müdahalesi ile yönü doğuya çevrilen robotun tekrar yönünü kuzeye çevirmesi gerektiği anlatılır. Öğrenciler gruplar hâlinde oluşturacakları programı ve kullanacakları program bloklarını tartışırlar. Rehber öğretmen gerekli noktalarda

yönlendirici sorular sormalı ve önerilerde bulunmalıdır. **Fakat çözümü öğrencilere hazır olarak vermemelidir.**

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

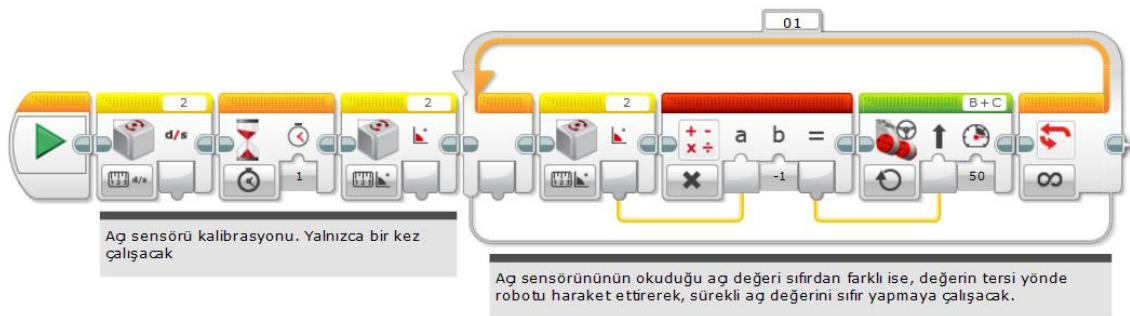
- Robot düz olarak gidip gitmediğini nasıl algılayabilir?
- Robot başlangıçtaki yönünden ne kadar saptığını nasıl algılayabilir?
- Robot rotadan çıkıp çıkmadığını ne zaman kontrol etmelidir?
- Robotun tekrar rotasına dönüşü nasıl sağlanabilir?

Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen işlemleri robotun nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdakilere benzer fikirler üretebilir:

- Düz gitmek için açı sensörü ve “döngü (loop)” bloğu beraber kullanılır.
- Rotadan sapma açısı hesaplanmalıdır.
- Tekrar rotaya dönmek için sapma açısı kadar dönüş yapılır.

3. ÜRET

Bu bölümde de öğrenciler aktif rol üstlenir. Rehber öğretmen yalnızca yönlendirir ve öğrencilere takıldıkları noktalarda destek olur. Yani, rehber öğretmen yalnızca ihtiyaç anında destek sağlamalıdır. Üret aşamasında, öğrencilerden bir önceki adımda tasarladıkları açı sensörü yardımıyla ilerleyen robotu kullanarak probleme daha yapılandırılmış bir çözüm önerisi geliştirmeleri istenir. Öğrenciler bilgisayar ve robot başında çalışarak gerekli yazılım çözümlerini geliştirir. Burada öğrencilerin en çok zorlanacağı konu, sapma açısının hesaplanarak aktarılmasıdır. Bu konuda öğrencilere yardımcı olunabilir. Örnek bir çözümün ekran görüntüsü aşağıdaki resimde verilmiştir.



Resim 77. Örnek Program

4. DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Bugün tasarladıkları “bulunduğu yerde dönen robot” uygulamasına farklı açılar girilerek robot döndürülebilir, robot durduğunda karşısına denk gelen öğrenciler seçilir ve aşağıdaki sorular üzerinden tartışma ortamı yaratılır:

- Açık sensör kullanmamaları gerekseydi bugün yapılan etkinlikleri şimdiye kadar gördükleri hangi araçlarla yapabilirlerdi?
- Açık sensör ile neler yapılabilir? Günlük hayatta ne gibi kullanım alanları olabilir? (Örneğin cep telefonlarında yatay veya dikey durumda olmasının algılanması)
- Açık sensörünü kullanarak robotun ilerlediği düzlemin eğimi ölçülebilir mi? Bunun için nasıl bir robot tasarımı yapılmalı?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilebilir.

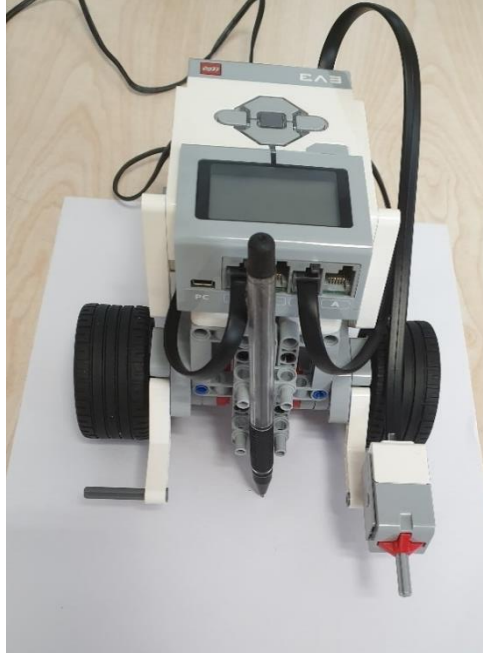
5. İLAVE ETKİNLİK

5.1. Kalemle Zemine Çokgen Çizdirme

Bu ilave etkinlikte robota bir kalem bağlandıktan sonra açık sensör yardımıyla hareket ettirilerek farklı şekiller çizdirilir. Öncelikle setteki parçalar kullanılarak robotun önüne aşağıdaki resimde gösterildiği gibi bir kalem tutturulur. Robotun A4 boyutunda bir kâğıdın üzerinde hareket ederken belirli geometrik şekilleri çizmesi istenir. Bu geometrik şekiller beşgen, altıgen, yedigen, sekizgen vb. olabilir.

Robotun istenilen çokgeni çizebilmesi için, ilgili çokgenin köşelerinde, çokgenin dış açısı kadar dönüş yapması gerekmektedir. Düzgün çokgenin bir dış açısı “ $360/\text{köşe}$ ” sayısı formülü ile hesaplanır.

Öğrencilerin çokgenler ve açılar konusunda yeterli ön bilgilerinin olmaması durumunda, dış açı değerleri rehber öğretmen tarafından verilebilir. Örneğin, robotun düzgün üçgen çizebilmesi için her köşede 120 derece, düzgün beşgen çizebilmesi için her köşede 72 derece dönmelidir.



Resim 78. Kalem Robotu Eklenmesi

5.2. Kalemle Zemine İstenilen Çokgeni Çizdirme

Etkinliklerin erken bitmesi durumunda, bir önceki etkinliğin geliştirilmesi istenebilir. Öğrencilerden robotun üzerinde bulunan düğmelerden yukarı düğmesine basıldığında üçgen, aşağı düğmesine basıldığında kare, sağ düğmesine basıldığında beşgen, sol düğmesine basıldığında altıgen çizen robotu programlamaları istenilir. Ayrıca çizim esnasında ekrana çizilen çokgenin ismi yazdırılabilir.

7. Hafta: Renk Sensörü

Ön bilgi:

- Öğrenciler temel düzeyde robot kavramını bilir.
- Öğrenciler robot setiyle farklı robot tasarımları yapmıştır.
- Öğrenciler robot setini programlamak için grafik arayüzünü kullanmıştır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler tuğlanın çeşitli sesleri çıkarması için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler tuğla ekranının görüntüsünü düzenlemek için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler dokunma, mesafe ve açılı sensörlerini farklı amaçlar için programlama adımlarını oluşturmuştur.

Haftanın Kazanımları:

- Öğrenciler renk sensörünün çalışma mantığını ve kullanma yöntemlerini açıklayabilir.
- Öğrenciler renk sensörünü farklı amaçlar için programlama adımlarını oluşturabilir.
- Öğrenciler robotun algıladığı renk, yansıyan ışık ve ortam ışığına göre davranması (hareket etmesi, ses çıkarması, tuğla ekranında farklı simgeler göstermesi) için gerekli programlama adımlarını oluşturabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin renk (colour) sensörünün çeşitli kullanım şekillerini (renk algılama, ortam ışığı ve yansıyan ışığını ölçme) kavrayarak sensörü çeşitli amaçlar için programlarken gerekli düzenlemeleri yapabilmesini sağlamaktır.

Kullanılacak Malzemeler:

Robot seti; bilgisayar; renk, açılı, dokunma ve mesafe sensörleri; mat (çalışma alanı).

Haftanın İşlenişi:

Gözele: Renk sensörünü kullanarak renkleri, yansıyan ışığı ve ortam ışığını algılama; yansıyan ışığı kullanarak robotu hareket ettirme (izleri / yolları takip etme); robotun ortam ışığına göre tepki vermesini (ışığı açma, yavaşlama, ses çıkarma) sağlama, renk sensörünü diğer sensörlerle birlikte kullanma.

Uygula: Her bir komutun çeşitli bileşenlerini programlayarak uygulama.

Tasarla: Robotun istenilen işlemleri yapabilmesi için gerekli bileşenleri tanımlama ve planlama.

Üret: Verilen görevleri programlama.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği.

1. GÖZLE VE UYGULA

1.1. Gözle: Renk Sensörü Nedir ve Nasıl Kullanılır?

Robotlar soyut olan programları, insanların yaşadığı somut fiziksel ortamlarda çalıştırır. Dolayısıyla, fiziksel dünya ile etkileşim hâlinde dirler. Fiziksel dünyada bir nesnenin rengi veya bir kaynaktan gelen ışığın şiddeti önemlidir. Örneğin, bir şoförün trafik ışıklarında ne yapması gerektiğini öğrenmesi için öncelikle renkleri algılayabilmesi gerekir. Başka bir örnek vermek gerekirse tablet, cep telefonu ve televizyon gibi cihazların ışık şiddeti uygun ayarlanmadığında bu cihazlardan gelen görüntünün algılanmasında zorluk yaşanabilir. Benzer şekilde robotlar için de renk çeşidi veya ışık şiddeti önemlidir. EV3 setinde bu iş için renk sensörü bulunur. Renk sensörü hem renk çeşidini hem de ışığın yoğunluğunu ölçmek için kullanılır. Renk sensörü “siyah” (1), “mavi” (2), “yeşil” (3), “sarı” (4), “kırmızı” (5), “beyaz” (6) ve “kahverengi” (7) renklerini algılayabilir. Bunun yanında karşısında hiçbir renk olmadığını da (0) algılayabilir. Burada renklerin yanına yazılan rakamlar, o renklerin kodlarıdır. Renk olmama durumu için “sıfır” kullanılır. Sensörün karşısında herhangi bir nesne yoksa sensör renk yok verisini iletir.



Resim 79. Renk Sensörü

Not

Renk sensörlü temel tasarıma (colour sensor down - driving base) ulaşmak için gerekli yönergeye aşağıdaki yollarla ulaşılabilir:

i) <https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-rem-color-sensor-down-driving-base-d30ed30610c3d6647d56e17bc64cf6e2.pdf>

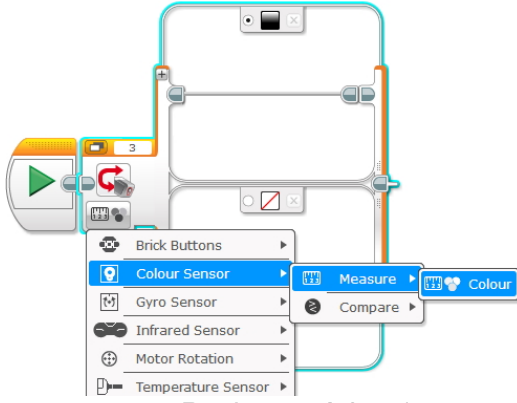
ii) EV3 yazılımı > Lobby > Building Instructions > Building Ideas > Color Sensor Down - Driving Base

iii) Robot setiyle birlikte gelen kitapçığa bakılabilir.

1.2. Gözle ve Uygula: Karşılaşılan Rengin İngilizcesini Söyleyen Program

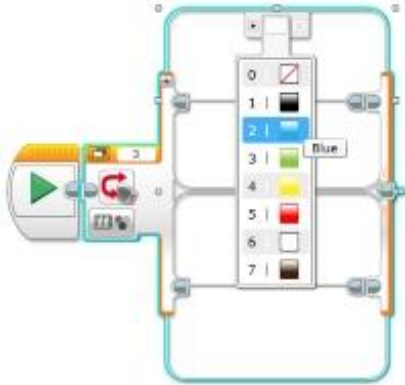
Öncelikle robotun renk sensörü takılır. Bu uygulamada, sabit duran bir robotun renk sensörüne gösterilen farklı cisimlerin renklerinin İngilizce adlarını söylemesini sağlayan bir program oluşturulacaktır. Bu programı oluşturmak için “döngü (loop)” bloğu içerisinde bir “anahtar (switch)” komutu kullanılmalıdır. Bu aşamada rehber öğretmen tarafından “anahtar (switch)” bloğunun ne işe yaradığı ve renk sensörü ile

nasıl kullanılabileceği kısaca anlatılır. Önceki haftalarda bahsedildiği gibi “anahtar (switch)” bloğu belirli durum veya koşullarda belirli işlemlerin yapılmasını sağlamak için kullanılır. Örneğin, robotun mavi rengi gördüğünde “blue” (mavi) demesi istenebilir. Burada bir koşul vardır: mavi olmak. Mavi olma koşulu sağlandığında akıllı tuğla “blue” diyen sesi oynatmalıdır. Aşağıda bu kodun nasıl oluşturulduğu adım adım gösterilmektedir.



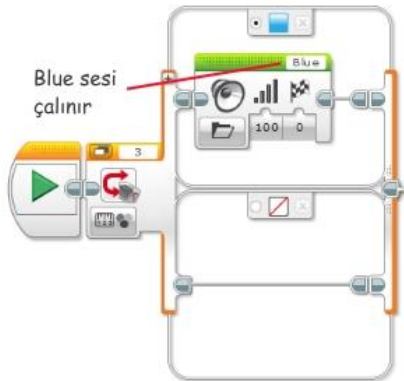
Resim 80. Adım 1

Anahtar (Switch) içerisinde Colour Sensor > Measure > Colour seçilir. Bunun anlamı koşul ifadesinde renk sensörünün renk karşılaştırmasına dayanan bir veya daha fazla durumun olacaktır.



Resim 81. Adım 2

Anahtar renklerinde “blue” (mavi) seçilir, yani mavi koşulu oluşturulur.

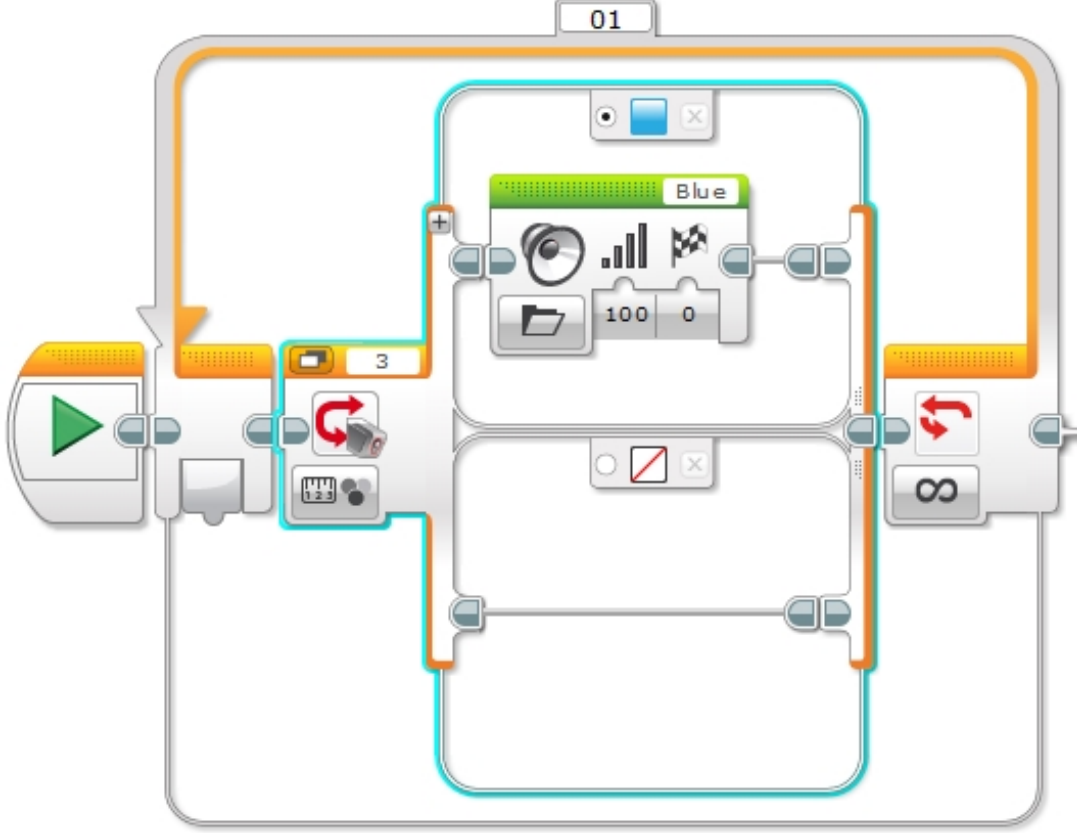


Resim 82. Adım 3

Cismin mavi renkli olması durumunda “blue” diyen ses oynatılır. Mavi olmadığı durumda ise herhangi bir şey yapılmaz.

Bu kodun eksik olduğu nokta, bu işlemi sadece program çalışır çalışmaz bir kere yapacak olmasıdır. Fakat robotun mavi rengi her gördüğünde mavi demesi

istenmektedir. Bu yüzden bu komutu bir döngü içerisine alıp sürekli tekrarlatmak gerekir. Bu programın son hâli aşağıdaki resimde gösterilmektedir.



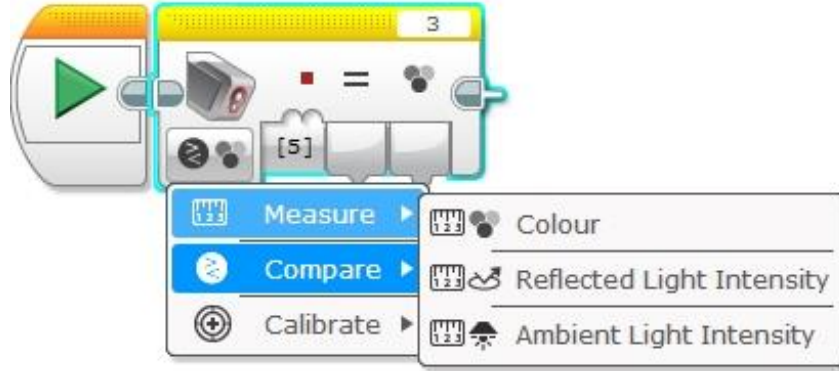
Resim 83. Örnek Program

Bu aşamada, öğrencilere “anahtar (switch)” ile birden fazla seçimin yapılabileceği gösterilerek ve uygulayarak anlatılır (**Hatırlatma:** Yeni koşullar eklemek için + işaretine (add case) tıklanır). Bundan sonra öğrencilerden tanımlı yedi rengin tamamının İngilizce adını söyleyen ve herhangi bir renk olmadığında “No” (Lego Sound Files> Communication klasörü içerisinde) diyen programı oluşturmaları istenir. Rehber öğretmen öğrencilere programı yazarken yardımcı olabilir fakat tam çözümü vermemelidir. (**Rehber öğretmen için ipucu:** Programı yazmak için + işaretine (add case) tıklanarak her renk için ve renk olmama durumu için birer koşul yaratılır ve gerekli ses oynatma komutları bu koşulların içerisine yazılır.) Öğrencilerin programı farklı renkler için çalıştırıp komutları denemesi sağlanmalıdır.

1.3. Gözle: Yansıyan Işık Şiddeti

Renk sensörü ışık şiddetini de ölçebilir. Işık şiddetini ölçmek için iki mod bulunur. Bunlar yansıyan ışık şiddeti ve ortam ışığı şiddetidir. Yansıyan ışık şiddetini ölçmek için renk sensörü karşısındaki yüzeye kırmızı bir ışık gönderir ve o yüzeyden yansıyan ışığın şiddetini “0” ile “100” arasında bir değer olarak belirler. “0” en karanlık, “100” ise en aydınlık değeri ifade eder. Ölçülen değer sıfıra ne kadar yakınsa yüzeyden yansıyan ışık o kadar azdır, yani karanlıktır. Ölçülen değer 100’e ne kadar yakınsa yüzeyden yansıyan ışık o kadar fazladır, yani aydınlıktır.

EV3 programında, sensor sekmesindeki “renk sensörü (colour sensor)” bloğu incelenerek renk sensörünün program içerisinde nasıl kullanılacağına bakılır. Renk sensöründe “Ölçüm (Measure)” ve “Karşılaştır (Compare)” olmak üzere iki seçenek bulunur. Ölçüm seçeneği, o sensörden ölçülen değeri hesaplar ve akıllı tuğlaya iletir. Örneğin, karşısında bir cisim bulunan renk sensörü, ölçüm seçeneği ile o cismin rengini (colour), o cisimden yansıyan ışığın yoğunluğunu (Reflected Light Intensity) ya da ortamda bulunan ışığın şiddetini (Ambient Light Intensity) hesaplayıp akıllı tuğlaya iletir. Karşılaştır (Compare) seçeneğinde ise ölçülen değer önceden belirlenmiş bir değer ile karşılaştırılır. Aşağıda resimde bu seçenekler görülmektedir.



Resim 84. Renk Sensörü Bloğu Seçenekleri

Bir cisimden yansıyan ışığın kırmızı olup olmadığını saptanması gerektiğini varsayalım. Bu durumda karşılaştırma seçeneği içerisinde “renk (colour)” kullanılarak cismin renginin kırmızı olup olmadığı bulunur. Aşağıdaki resimde görüldüğü gibi, karşılaştırma seçeneğinde karşılaştırılacak değer 5, yani kırmızı olarak belirlenir. Eğer karşıdaki cisim kırmızı ise doğru değeri, kırmızı değilse yanlış değeri üretilir.



Resim 85. Adım 1

Şimdi de bir cisimden yansıyan ışığın miktarının 80'den fazla olup olmadığını saptanması gerektiğini varsayalım. Karşılaştırma seçeneği içerisinde “yansıyan ışık yoğunluğu (reflected light intensity)” kullanılarak bu işlem gerçekleştirilebilir. Aşağıdaki resimde verilen komut bu işlem için kullanılır. Görüldüğü üzere büyüktür (>), büyük eşittir (≥), küçüktür (<), küçük eşittir (≤), eşittir (=), eşit değildir (≠) gibi karşılaştırma işlemleri gerçekleştirilebilir. Karşılaştırılmak istenen değer belirlenip yazılır. Örnekte bu

değerin “80” olarak belirlendiği görülebilir. Sonuçta yansıyan ışık 80’den büyükse, yani yeteri kadar aydınlıksa doğru değeri, değilse yanlış değeri üretilir.



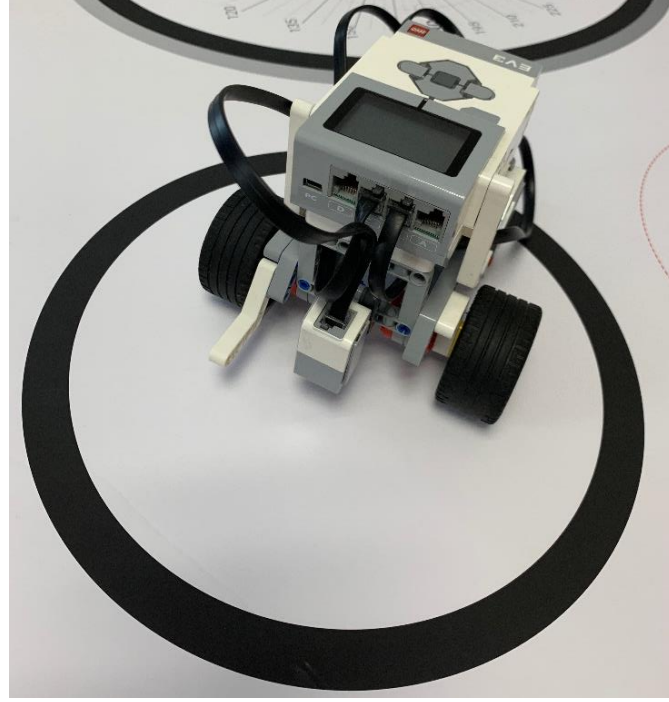
Resim 86. Adım 2

Dikkat:

Renk sensörü ile yapılan etkinliklerde (bu ve bundan sonraki haftalar dâhil olmak üzere) renk sensörünün rengi algılanacak olan cisme uzaklığı, cismin renginin yapısı ve ortamdaki ışık miktarı gibi parametreler programın çalışmasını etkileyebilir. Yazılan program(lar) çalışmıyor ise renk sensörünün konumu, kullanılan cisim ve ışık miktarı gibi parametreler rehber öğretmen tarafından düzenlenerek programın çalışması sağlanmalıdır.

1.4. Uygula: Dairenin İçerisinden Çıkmayan Robot

Bu etkinlikte, beyaz bir zemin üzerinde, siyah şeritle çizilmiş dairesel bir bölgede hareket edip bu bölgeden çıkmayan robot yapılır.



Resim 87. Kullanılacak Mat Bölgesi

Bu programın algoritması şöyledir:

- i. Siyah şeridi görünceye kadar ilerle.
- ii. Siyah şeridi görünce robotun sol tekerini 100 derece geriye çevir.
- iii. i ve ii numaralı adımları sürekli tekrarla.

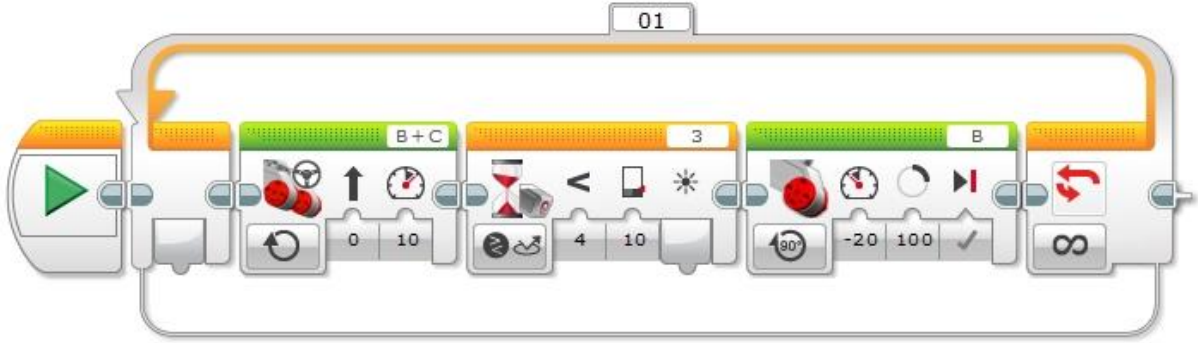
Bu program “anahtar (switch)” bloğu ve “bekle (wait)” bloğu kullanılarak iki farklı şekilde yapılır. Fakat öncelikle siyah şeride gelindiğini algılamak için kullanılması gereken eşik değerinin nasıl bulunacağından söz edilmelidir. Farklı yüzeylerde ve farklı renk tonlarında yansıyan ışık miktarı değişiklik gösterebilir. Bu yüzden her bir grup bu işlemi ayrı ayrı yapıp eşik değerini kendilerine göre belirlemelidir.

Eşik değerinin bulunması için aşağıdaki adımlar uygulanmalıdır:

- i. Renk sensörü beyaz yüzey üzerine getirilir.
- ii. Akıllı tuğla üzerinden veya robota bağlı bilgisayar üzerinden yansıyan ışık miktarı ölçülür.
- iii. Renk sensörü siyah yüzey üzerine getirilir.
- iv. Akıllı tuğla üzerinden veya robota bağlı bilgisayar üzerinden yansıyan ışık miktarı ölçülür.
- v. Eşik değeri bu iki değer arasından siyah yüzey üzerindeki değere yakın olarak seçilir.

Örneğin, beyaz yüzey için “20”, siyah yüzey için “7” değeri bulunmuş olsun. Bu durumda eşik değeri “10” olarak belirlenebilir. Eşik değeri daha önce de açıklandığı gibi farklı ortamlarda farklı değerler olarak bulunabilir/belirlenebilir. Bu değer kullanılarak sensörün siyah şeridin üzerine geldiği bilgisi robota iletilir. Yansıyan ışık miktarı 10’dan küçük olduğunda, robot siyah şeridin üzerinde olduğunu algılar.

Programı oluşturmak için öncelikle “bekle (wait)” bloğu kullanılır. Sol teker motorunun “B” portuna, sağ teker motorunun ise “C” portuna takıldığı varsayalım. Yapılması gereken işlem, renk sensöründen yansıyan değer, eşik değeri olarak belirlenen 10’un altına düşüncüye kadar (yani beyaz yüzeyden siyah yüzeye geçme işlemi kesinleşinceye kadar) robotu sürekli “10” hızında ilerletmektir. Eşik değerine geldikten sonra ise sol teker geriye doğru “20” hızında “100” derece döner (**Dikkat:** Burada robot “100” derece dönmez, yalnızca bir tekerlek “100” derecelik bir tur atar). İşlem bu hâliyle sadece bir kere yapılır. Fakat bu işlemin sürekli tekrarlanması gerekir. Bu yüzden de işlemlerin bir döngü içerisine alınması gerekmektedir. Sonuç olarak aşağıdaki resimde verilen örnekteki gibi bir program elde edilir.

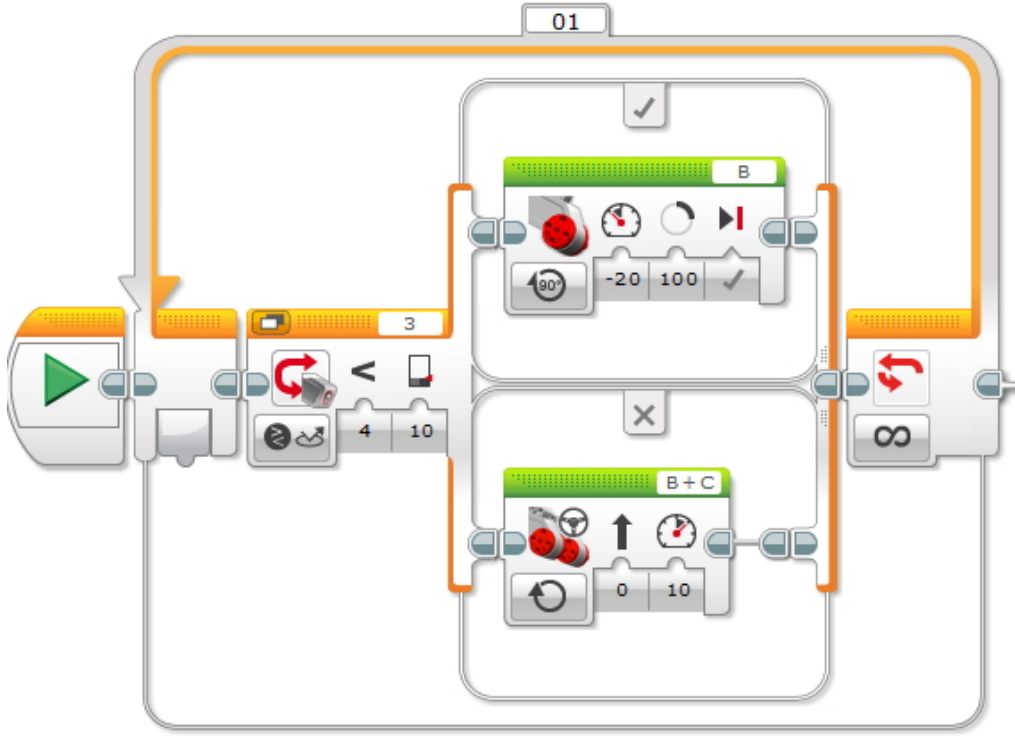


Resim 88. Örnek Program

Aynı görev, “bekle (wait)” bloğu kullanmadan “anahtar (switch)” bloğu ile yapılacak olursa mantığı şu şekilde kurgulanır:

- i. Ortamdan yansıyan ışığın değeri, eşik değerinin altına düşerse sol teker ile geriye git.
- ii. Ortamdan yansıyan ışığın değeri eşik değerinin üstündeyse ileriye git.
- iii. i ve ii numaralı işlemleri sürekli tekrarla.

Bu program adım adım oluşturularak öğrencilere detaylıca anlatılır ve son hâli aşağıdaki resimde olduğu gibi görünür.



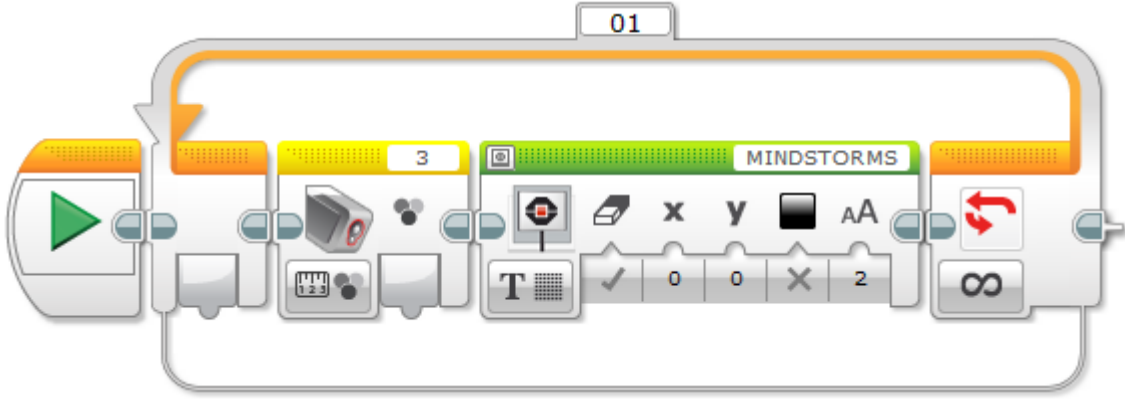
Resim 89. Örnek Program

1.5. Gözle: Ortam Işıđı

Ortam ışığı yoğunluğu modunda renk sensörü herhangi bir ışık göndermez. Doğrudan bulunduğu ortamdan gelen ışığın yoğunluğu ile ilgilenir. Bu yoğunluk değeri yine "0" ile "100" arasındadır. Sıfır en karanlık, "100" ise en aydınlık değeri ifade eder. Ortamda bulunan ışığın şiddeti (ambient light intensity) hesaplanıp akıllı tuğlaya iletilir.

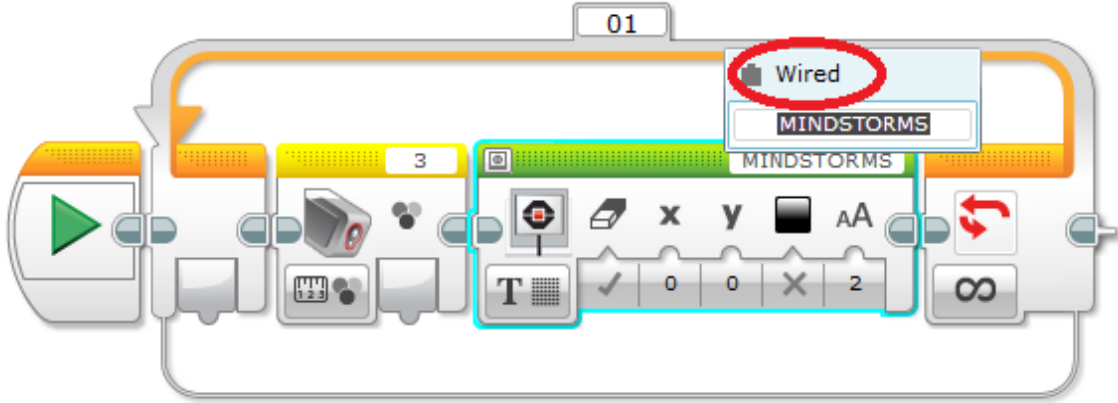
1.6. Uygula: Ortam Işıđı

Bu etkinlikte renk sensörünün ölçtüđü ortam ışığı değeri her iki saniyede bir ekrana yazdıran program oluşturulur. Ortam ışığı her iki saniyede bir ölçülmek istendiđi için öncelikle renk sensörü blođu bir döngünün içine alınır. Sensörün ölçtüđü değeri ekrana yazdırmak için "ekran (display)" blođu renk sensörüne bağlanıp aşğıdaki resimde olduđu gibi "metin (text)" seçeneđi aktif hâle getirilmelidir.



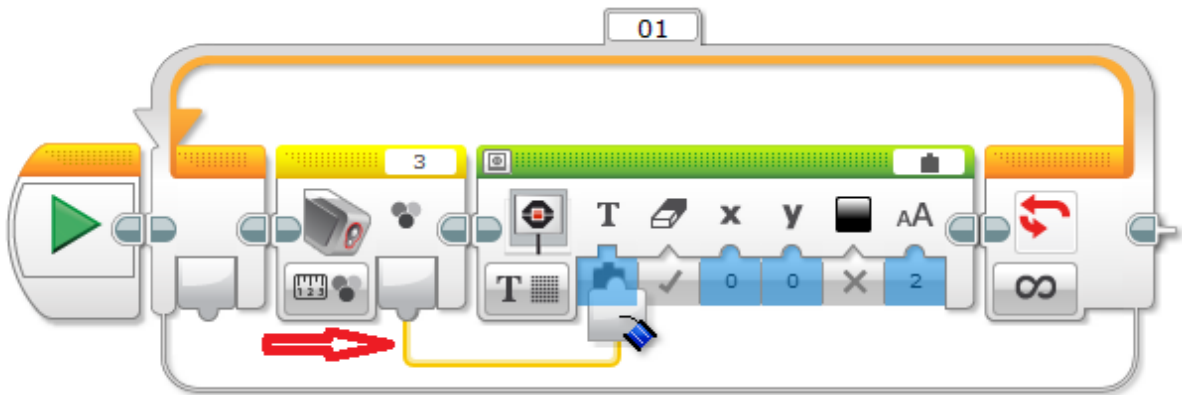
Resim 90. Adım 1

Renk sensörünün ölçtüğü değeri tuğlanın ekranında yazdırmak için öncelikle display bloğundaki içerik seçeneği aşağıdaki resimde olduğu gibi “Kablolu (Wired)” olarak belirlenir.



Resim 91. Adım 2

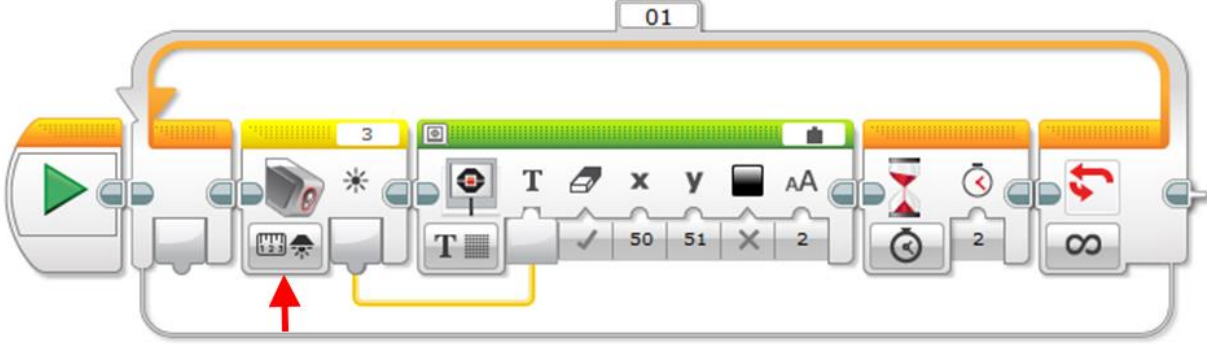
Sonra da renk sensörü bloğunun ölçtüğü değeri tuğla ekranında göstermek için renk sensörü bloğu aşağıdaki resimde olduğu gibi “ekran (display)” bloğu ile bağlanır.



Resim 92. Adım 3

Belirlenen değerin iki saniyede bir ölçülmesi için ise aşağıdaki resimde olduğu gibi “ekran (display)” bloğunun sonuna “bekle (wait)” bloğu eklenip bekleme süresi 2 yapılır.

Son olarak renk sensörünün ortam ışığını ölçmesi için “Ölçüm (Measure)” menüsü içerisinde “Ortam Işığı Şiddeti (Ambient Light Intensity)” seçilir.

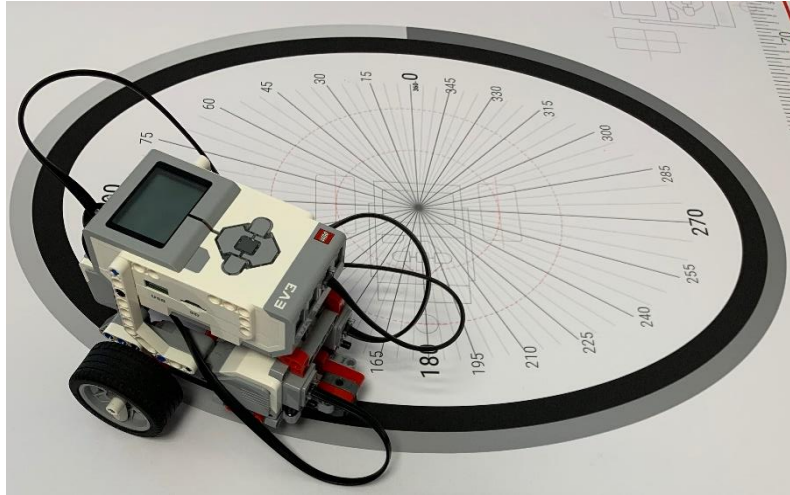


Resim 93. Adım 4

2. TASARLA VE ÜRET

2.1. Tasarla: Eliptik Bir Yörüngeyi Takip Eden Robot

Bu etkinlikte, beyaz bir zemin üzerinde, siyah bir şeritle elips şeklinde çizilmiş bir yörüngeyi sürekli takip edecek bir robotun programı hazırlanır. Program için sensörün elipsin iç tarafında olduğu ve robotun saat yönünün tersine doğru ilerlediği varsayılır. (**Dikkat:** Aksi hâlde programın çalışmayacağı vurgulanmalıdır). Tasarlama sürecinde öğrencilerin eliptik bir yörüngeyi takip eden robot için aşağıda örnek olarak verilen iki adıma benzer bir süreci gerçekleştirmesi gerekir. Bu etkinlikte mat üzerinde ilgili bölüm kullanılır.



Resim 94. Kullanılacak Mat Bölgesi

Öğrencilerden öncelikle beyaz bir zemin üzerinde, siyah bir şeritle elips şeklinde çizilmiş bir yörüngeyi sürekli takip edecek bir robotun program kodunun nasıl oluşturulacağı üzerine düşünceleri istenir. Öğrenciler grup olarak tartışır. Rehber

öğretmen onlara gerektiği noktada yardımcı olabilir fakat tam bir çözüm vermemelidir, gruplar çözümlerini kendileri üretmelidir.

Tanımlama: Öğrencilerin öncelikle istenilen işlemin neler gerektirdiğini belirlemesi ve maddeler hâlinde yazması gerekir. Örneğin:

- i. Robot sürekli olarak hareket edecek.
- ii. Robot siyah bölgeye geldiğinde beyaz bölgeye yönlendirilecek.
- iii. Robot beyaz bölgeye geldiğinde siyah bölgeye yönlendirilecek.
- iv. Robot durduruluncaya kadar bu işleme devam edecek.

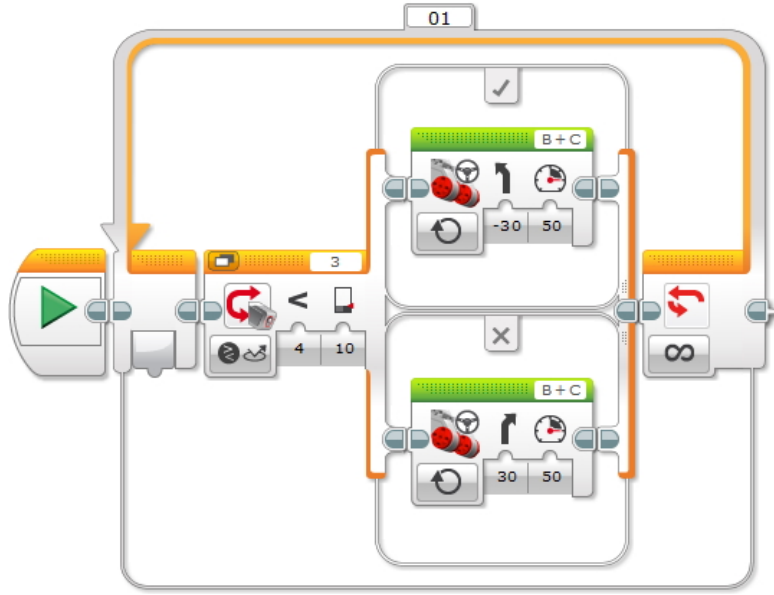
Fikir üretme: Bu aşamada öğrencilerin yukarıda belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi gerekir. Örneğin, öğrenciler aşağıdaki maddelere benzer fikirler üretilebilir.

- i. Robotun siyah çizgiye geldiğinin anlaşılması için eşik değeri belirlenir (beyaz bölgeden siyah bölgeye geçtiğini anlamak için).
- ii. Robot ileri doğru belirli bir hızla hareket eder.
- iii. Robotun beyaz bölgeden çıkma durumu sürekli olarak kontrol edilir (renk sensöründen gelen verinin eşik değerine yaklaşma durumu sürekli kontrol edilir).
- iv. Robot siyah bölgeye geldiğinde dairenin içerisine doğru (dönüp geri gidecek şekilde) yönlendirilir.
- v. ii, iii ve iv numaralı adımlar robot durduruluncaya kadar tekrarlanır.

Bu programın temel mantığı robot siyah bölgeye geldiğinde onu beyaz bölgeye yönlendirmek, beyaz bölgeye geldiğindeyse onu siyah bölgeye yönlendirmektir. Böylece robot zikzaklar çizerek ilerler. Bir önceki etkinlikte olduğu gibi, robotun siyah bölgede olduğunu anlaması için bir eşik değeri belirlenmelidir. Bunun için her bir grubun kendi programlarında kendi buldukları eşik değerini kullanmaları gerekir.

2.2. Üret: Eliptik Bir Yörüngeyi Takip Eden Robot

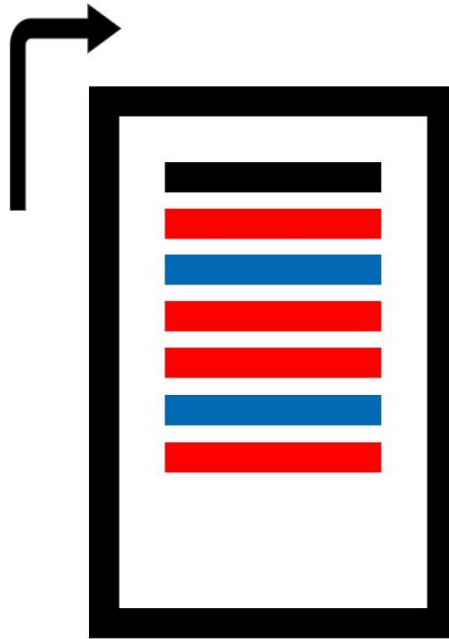
Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenilen görevi yerine getirir. Öğrencilerin aşağıdaki resimde görülen programa benzer bir program hazırlamaları beklenmektedir.



Resim 95. Örnek Program

2.3. Tasarla: Dikdörtgen Üzerinde Hareket Eden Robot

Bu etkinlikteki amaç, robotun dikdörtgen şeklindeki bir çizgiyi takip etmesi için program oluşturmaktır. Mat üzerinde görüldüğü gibi, dikdörtgenin kenarları siyah bir şeritle çizilmiş ve arka plan rengi beyaz olarak tasarlanmıştır. Program için robotun dikdörtgenin kenarlarının dış kısmını (iç kısımdan takip eden de yapılabilir fakat burada dış kısımdan takip etmelidir) takip ederek saat yönünde ilerlediği varsayılır. Aşağıda resimde dikdörtgen örneği görülmektedir.

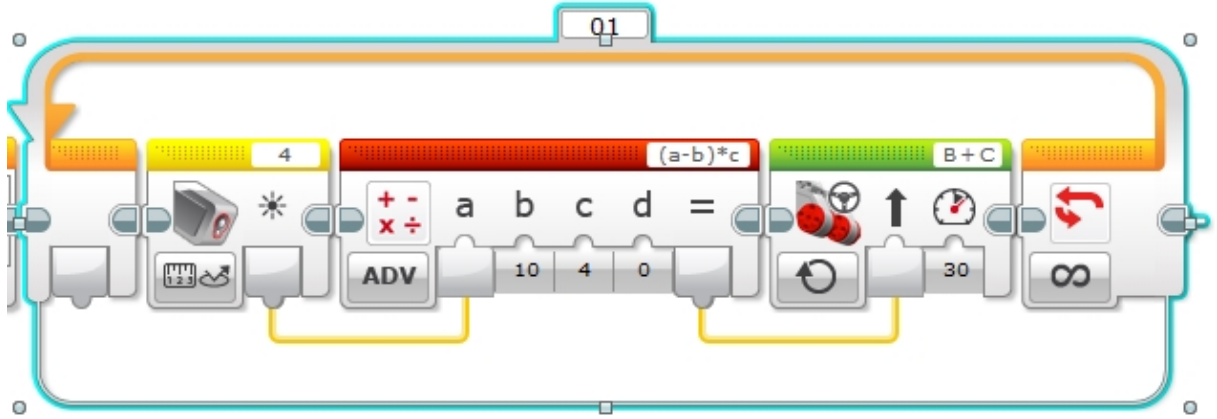


Resim 96. Dikdörtgen Takip Bölgesi

Robotun dikdörtgenin dış sınır çizgisini (siyah çizginin kendisini değil veya orta noktasını değil, siyah çizgi ile beyaz zeminin kesiştiği hattı) takip etmesi beklenir. Robotun sınır çizgisini birebir takip etmesi zorunlu olmasa da bu çizgiye yakın bir şekilde hareket etmesi gerekir. Programı yazmaya başlamadan önce grupların tasarla adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmesi gerekir.

2.4. Üret: Dikdörtgen Üzerinde Hareket Eden Robot

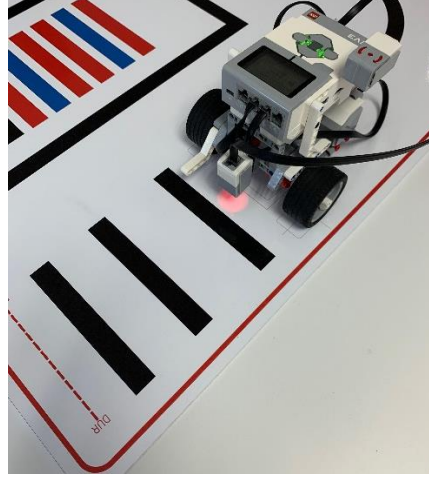
Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenilen görevi yerine getirmelidir. Aşağıdaki resimde görülen çözüme benzer bir örnek hazırlanabilir. Aşağıda verilen örnek çözüm öğrencilere gösterilmemelidir. Üret adımlarının tamamında öğrenciler kendi çözümlerini kendileri üretmelidir. Aşağıdaki çözümde dikdörtgenden yansıyan ışık miktarı değeri (b) "10" olarak belirlenmiştir. Fakat bu değer ortama göre farklılık gösterebilir. Öğrenciler robotu kullanarak bu değeri kendi ölçümlerine göre belirlemelidir. Örnek programdaki "c" değeri, robotun çizgiden uzaklaştığında yeniden çizgiye yönelmesi için gerekli reaksiyon hızı katsayısıdır. Bu değer ne kadar fazla olursa robot sapmaya o kadar hızlı tepki verir. Fakat bu değer de programcı tarafından kendi durumuna göre ayarlanmalıdır. "C" değerinin fazla veya az olması robotun çizgiyi takip etmemesine yol açabilir. Son olarak, robot dikdörtgenin dış yüzünde olmalıdır. Eğer iç yüzüne konulursa program çalışmaz.



Resim 97. Örnek Program

2.5. Tasarla: Üçüncü Çizgiyi Geçince Duran Robot

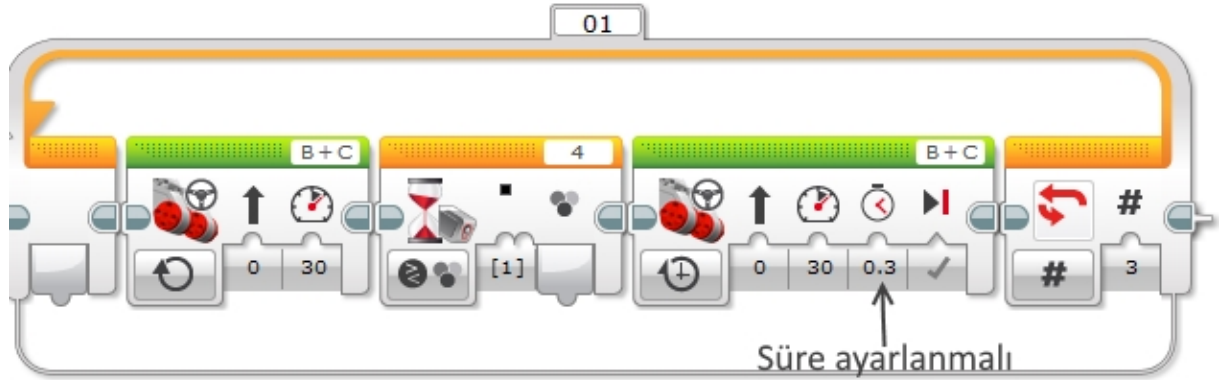
Bu etkinlikteki amaç, robotun aşağıdaki resimde gösterildiği gibi üçüncü çizgiyi kısa bir miktar geçince durması için gereken programı oluşturmaktır. Robot ilk iki çizgiyi tamamen geçtikten sonra üçüncü çizgiyi biraz geçip durmalıdır. Programı yazmaya başlamadan önce grupların tasarlama adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme sürecini gerçekleştirmesi gerekir. Rehber öğretmen öğrencilerin tanımlama ve fikir üretme adımlarını gerçekleştirmelerini sağlamalıdır. Burada amaç problemi çözmeden önce öğrencilerin problemin ve çözümünün üzerine düşünmelerini sağlamaktır.



Resim 98. Kullanılacak Mat Bölgesi

2.6. Üret: Üçüncü Çizgiyi Geçince Duran Robot

Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenilen görevi yerine getirir. Aşağıdaki resimde görülen örnek çözüme benzer bir program hazırlanabilir. Döngünün içerisindeki ikinci “direksiyon hareketi (move steering)” bloğunun süresi çizgilerin kalınlığına göre ayarlanmalıdır.



Resim 99. Örnek Program

3. DEĞERLENDİR

Bu aşamada hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşüncelerini sağlamaktır. Bu sayede öğrenciler; problem çözme yetenekleri, dersin konusu ve kendileri ile ilgili gözlemler yaparak öğrendikleri yeni konuları ve kendilerini değerlendirmekle beraber sonraki çalışmalarını planlamak için de fırsat elde edeceklerdir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Verilen görevleri göz önünde bulundurduğunuzda en çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
 - Eliptik yörüngeyi takip eden robotta, robotun çizgide kalmasını nasıl sağladınız?

- Dikdörtgen üzerinde hareket eden robotta, robotun köşeyi dönmesini nasıl sağladınız?
- Üç çizgi etkinliğinde, robotun siyah alanları geçmesini nasıl sağladınız?
- Kullandığınız yöntemler bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla anlaşmazlığa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne/neler öğrendiniz?
- Bu hafta öğrendiğiniz bilgileri günlük hayata uygulamak isteseniz, hangi sorunların çözümünde bu bilgileri kullanabilirsiniz?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilebilir.

8. Hafta: Sensörlerin Birlikte Kullanımı

Ön bilgi:

- Öğrenciler robot setiyle farklı robot tasarımları yapmıştır.
- Öğrenciler robot setini programlamak için grafik arayüzünü kullanmıştır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler akıllı tuğlanın çeşitli sesleri çıkarması için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler robot ekranını kullanmak için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler robotu programlarken “döngü (loop)” ve “anahtar (switch)” bloklarını kullanmıştır.
- Öğrenciler sensörlerin özelliklerini kullanarak program kurgulamıştır.

Haftanın Kazanımları:

- Öğrenciler farklı sensörleri bir arada kullanır.
- Öğrenciler farklı sensörlerden alınan verileri işler.
- Öğrenciler ileri matematik ve mantık işlemlerini kullanır.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin EV3 robot setinde bulunan farklı sensörleri bir arada kullanabilmesini ve farklı sensörlerden elde edilen veriler üzerinde matematik ve mantık işlemleri gerçekleştirebilmesini sağlamaktır.

Kullanılacak Malzemeler:

Robot seti, bilgisayar, mat (çalışma alanı)

Haftanın İşlenişi:

Gözle: Farklı sensörleri bir arada kullanma ve matematik ve mantık işlemlerini gerçekleştirme

Uygula: Farklı sensörleri kullanarak çeşitli uygulamalar geliştirme

Tasarla: Sensörleri kullanarak çizgi izleyen ve engelde duran robot tasarımı için gerekli bileşenleri tanımlama ve planlama

Üret: Robotu verilen görevlere göre programlama

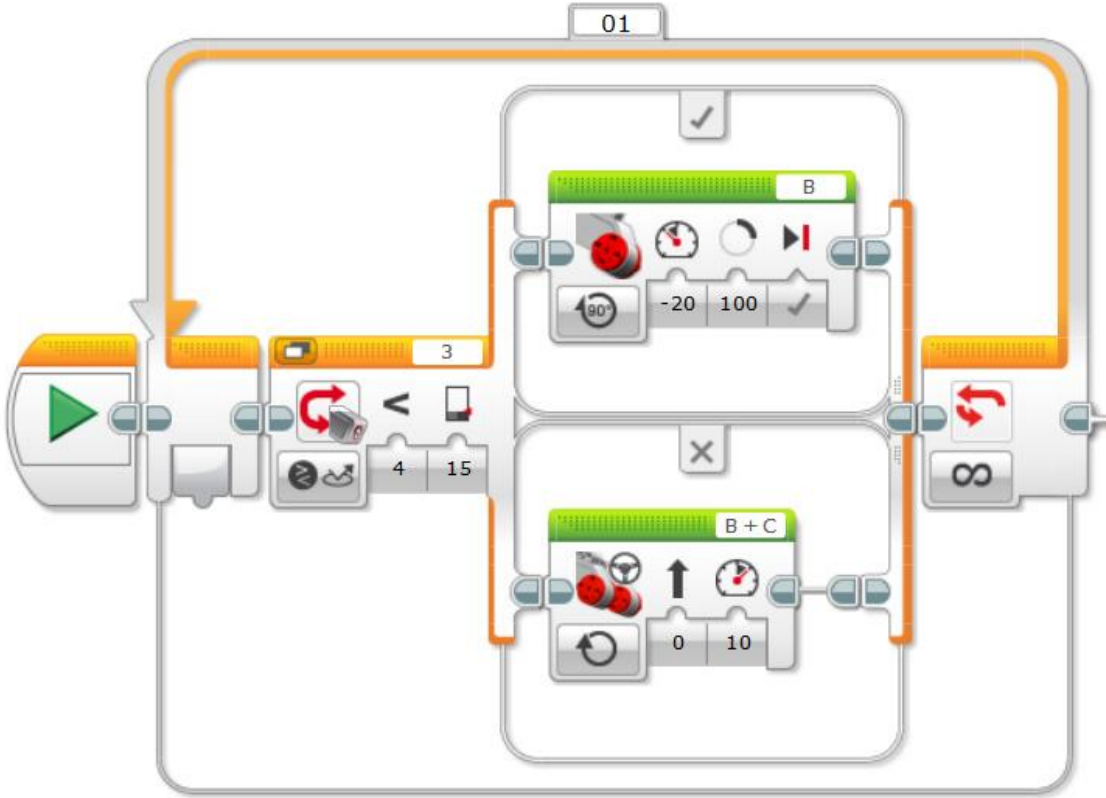
Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği

1. GÖZLE VE UYGULA

1.1. Gözle: Farklı sensörleri bir arada kullanma

Öğrencilere önceki haftalarda kullandıkları anahtar (switch) ve döngü (loop) kavramları tekrar edilir. Birden fazla sensörün eş zamanlı olarak kullanıp kullanılmayacağı tartışılır. Sensörleri eş zamanlı kullanılabilecekleri görevlere örnekler vermeleri istenir. Örneğin, düz ilerlerken engele olan uzaklığı 20 cm'nin altında ise yavaşlayan ve engele dokununca duran robot veya siyah renkle belirtilmiş alandan dışarı çıkmayan ve alan içerisindeki bir engele de çarpmayan robot.

Öğrencilerin verdiği örneklerden uygun bulunan birinin veya yukarıda verilen örneklerden birinin nasıl gerçekleştirileceği gösterilir. Bu uygulamalar için birbiri içerisinde yer alan anahtar blokları kullanılır. Yukarıdaki örneklerden, belirlenen alandan dışarı çıkmayan ve alan içerisinde yer alan engele de çarpmayan robot için şöyle bir yol izlenir (Robota mesafe ve renk sensörleri takılı olmalıdır). Renk sensörü haftasında oluşturulan, daire içerisinde çıkmayan robot programı öğrencilere hatırlatılır. Örnek kodun bir bölümü aşağıdaki resimde görülmektedir.

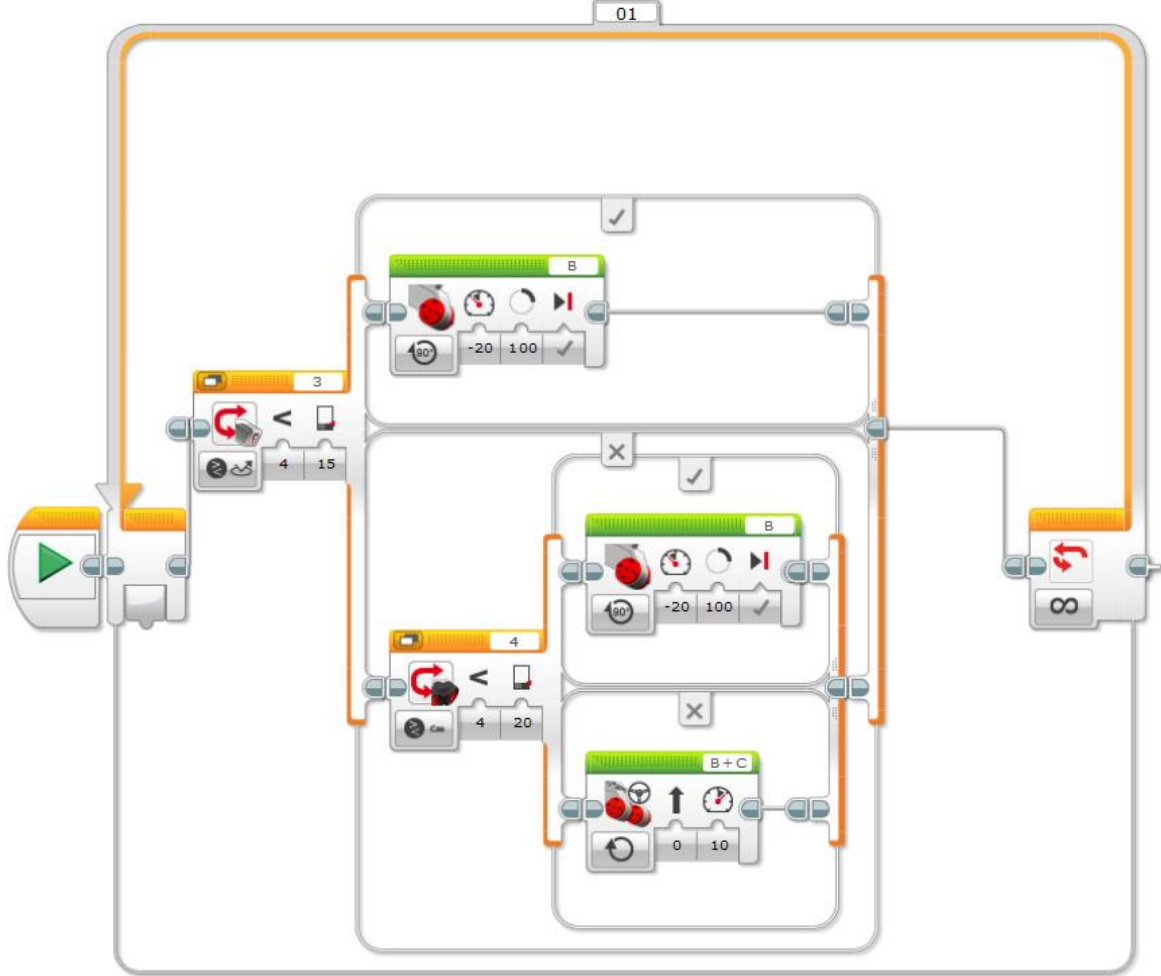


Resim 100. Örnek Program

Bu programda renk sensörünün ölçtüğü yansıyan ışık miktarına göre robot düz gider ya da dönme hareketi gerçekleştirir. Yansıyan ışık miktarı 15'in altındaysa (yani robot siyah alana gelmişse) robot döner, yansıyan ışık miktarı 15'in üstündeyse düz gitmeye devam eder (**Dikkat:** 15 değeri ortama göre değişebilir, robot siyah alanı algılamıyorsa 10 veya daha küçük değerler denenir).

Robot düz giderken karşısında bir engel olup olmadığını anlayabilmesi için ise mesafe sensörüne ihtiyacı vardır. Karşısında bir engel yoksa robot düz gitmeye devam eder, engel olması durumunda ise yönünü değiştirir. Var olan programa bir “anahtar (switch)” bloğu daha ekleyerek düz gitme durumuna bir koşul daha tanımlanır. Böylece robot düz giderken mesafe sensöründen algıladığı veriyi de değerlendirir ve duruma göre düz gider veya dönme hareketini gerçekleştirir.

Programın son hâli aşağıdaki resimde görülmektedir. **Bu program adım adım oluşturulmalı ve öğrencilere detaylıca anlatılmalıdır.**

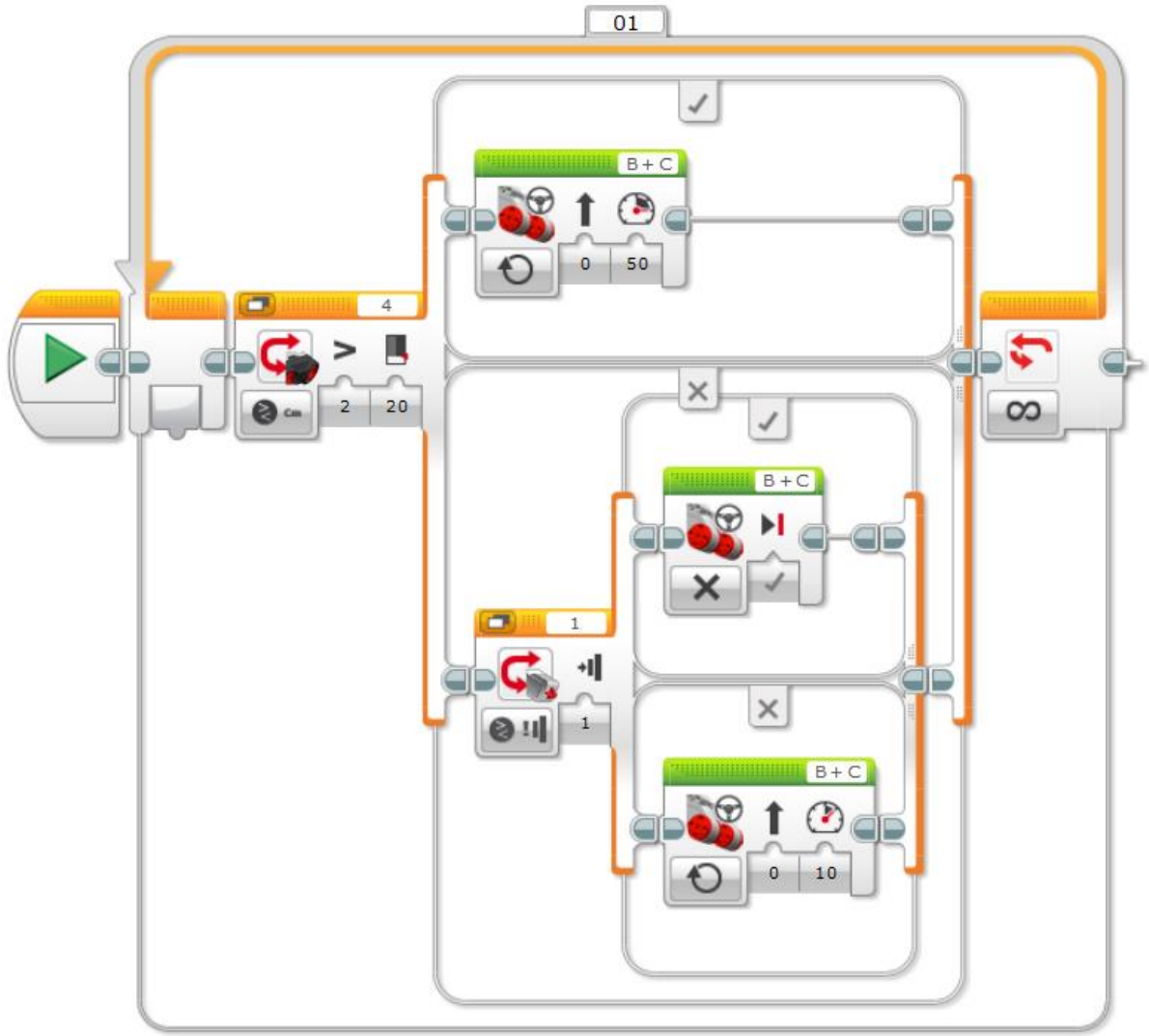


Resim 101. Örnek Program

1.2. Uygula: Farklı sensörleri bir arada kullanma

Farklı sensör kombinasyonları ile çok farklı uygulamalar gerçekleştirilebilir. Bu aşamada, öğrencilerin önerdiği örneklerden biri rehber öğretmen tarafından seçilir veya bir engele olan uzaklığı 20 cm'nin altında ise yavaşlayan ve engele dokununca duran robotun programlanması istenir.

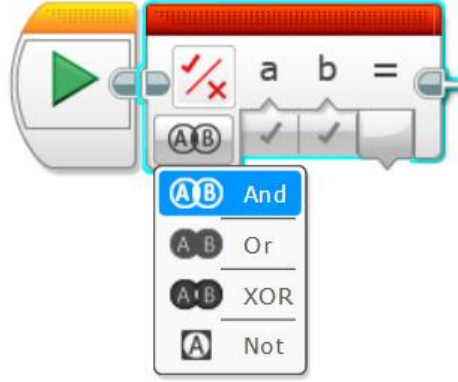
Aşağıdaki resimde görülen örnek program incelenebilir.







Resim 102. Örnek Program

1.3. Gözle: Mantık İşlemi (Logic Operation) Bloğu

Yalnızca belirli bir koşulun gerçekleşmesi durumunda yapılması gereken görevler için “mantık işlemi (logic operation)” bloğu da kullanılabilir. “Logic operation” bloğu, Veri İşlemleri (Data Operations) sekmesi altında yer alır. “Mantık işlemi (Logic operation)” bloğu ve aldığı parametreler ilgili açıklamalar aşağıdaki resimde görülmektedir.



Resim 103. Mantık İşlemi / Logic Operation Bloğu

İkon	Mod	Girdiler	Çıktılar
 And	AND	A, B	A ve B'nin her ikisi de doğruysa sonuç doğru, aksi takdirde sonuç yanlış.
 Or	OR	A, B	A veya B (veya her ikisi de) doğru ise sonuç doğru, hem A hem de B yanlış ise sonuç yanlış.
 XOR	XOR	A, B	Yalnızca A ve B'nin biri (ve yalnızca biri) doğru ise sonuç doğru. A ve B (her ikisi de) doğruysa sonuç yanlış. Hem A hem de B (her ikisi de) yanlışsa da sonuç yanlış.
 Not	NOT	A	Sonuç girilen değer tersidir. A yanlış ise sonuç doğrudur. A doğru ise sonuç yanlıştır.

Ortaokul öğrencileri mantık işlemlerini anlamakta zorlanabilirler. Öğrencilere sadece “VE (AND)” ve “VEYA (OR)” işlemlerinin anlatılması yeterli olacaktır. Öğrencilerin daha kolay anlamalarını sağlamak amacıyla günlük hayattan örnekler verilebilir. Örneğin yapılacak olan bir salatada “havuç” VE “salatalık” olsun ifadesi ile “havuç” VEYA “salatalık” olsun ifadeleri arasındaki anlam farklılıkları tartışılabilir. Ayrıca bir internet arama motorunda “kedi” AND “köpek” şeklinde yapılan bir arama ile “kedi” OR “köpek” şeklinde yapılan bir aramada bulunan sonuç sayıları kıyaslanabilir. Ayrıca arama motorunun bulunduğu sayfaların içerikleri tartışılabilir. Benzer örneklerle öğrencilerin “AND” ve “OR” işlemlerini kavramları sağlanır.

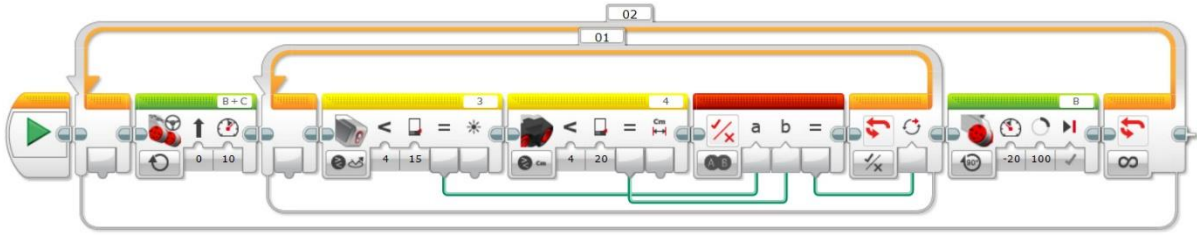
1.4. Uygula: OR (Veya) Mantık İşlemi

Öğrencilerden gün içerisinde yaptıkları, siyah renkle belirtilmiş alanın dışına çıkmayan ve alan içerisindeki bir engelle de çarpmayan robotu “mantık işlemi (logic operation)” bloğunu kullanarak tasarlamaları istenir.

Görevi gerçekleştiren programı oluşturabilmeleri için öğrencilere yönlendirici sorularla rehberlik edilir. Örneğin:

- Robot hangi koşullarda dönme hareketi gerçekleştirir?
- Hangi mantık işlemi (And, Or, Xor, Not) kullanılmalı? Neden?

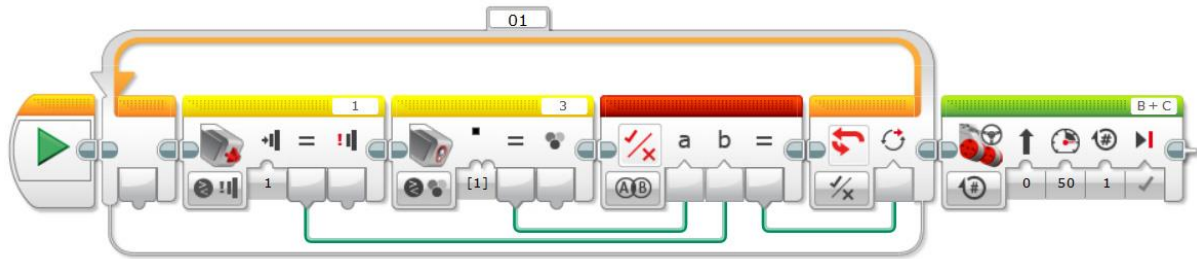
Örnek program aşağıdaki resimde görülmektedir.



Resim 104. Örnek Program

1.5. Uygula: AND (Ve) Mantık İşlemi

Robotun belirlenen koşulların hepsinin sağlanması durumunda bir görevi gerçekleştirmesi istendiğinde hangi mantık işleminin kullanılması gerektiği öğrencilerle tartışılır. Benzer şekilde öğrencilerle yeni görevler kurgulanır. Öğrencilerin kurguladığı görevlerden biri seçilerek tüm grupların aynı görevi gerçekleştirmesi istenir. Örneğin, siyah zemin üzerinde bulunduğu ve dokunma sensörü tetiklendiği zaman harekete geçen, koşullar sağlanmamış ise harekete geçmeyen bir robot için program oluşturulabilir. Bunun için aşağıdaki resimde görülen örnek program kullanılabilir.



Resim 105. Örnek Program

2. TASARLA VE ÜRET

2.1. Tasarla: Çizgi İzleyen ve Engelde Duran Robot

Bu etkinliğin amacı, öğrencilerin önceki hafta yaptıkları çizgi izleyen robot programına mesafe sensörü de ekleyerek çizgi izleme esnasında karşısına bir engel çıktığında robotun durmasını, engel kaldırıldığında çizgiyi takip etmeye devam etmesini sağlayan programı kurgulamaktır.

Öğrencilerle çizgi izleyen robot programı ve kullanılan renk sensörü tartışılır. Renk sensörünün algıladığı siyah ve beyaz renklere göre nasıl hareket ettiğiyle beraber programda kullanılan switch bloğu ve loop bloğunun nasıl çalıştığı da tekrar edilir. Öğrencilerden robotun çizgi üzerinde bir engel ile karşılaşması durumunda bunu bir ses ile bildirmesi ve beklemesini sağlayacak, engel ortadan kalkınca da hareketine devam edecek şekilde robotları programlamaları istenir. Öğrenciler, grup olarak kurgulanacak programı ve kullanılacak program bloklarını tartışırlar.

Tanımlama: Öğrenciler öncelikle problemi tanımlayabilmelidir. Problemi çözmelerine yardımcı olacak aşağıdaki soruların cevaplarını kendi aralarında tartışmalıdırlar.

- Robot çizgiyi takip ederken bir engel ile karşılaştığını nasıl algılayabilir?
- Engel ile karşılaştığında robotun çizgi izleme hareketi nasıl durdurulabilir?
- Robot öncelikle engeli mi yoksa çizgiyi mi algılamalı?
- Engel ortadan kalkınca robot başlangıçtaki görevine nasıl devam edebilir?

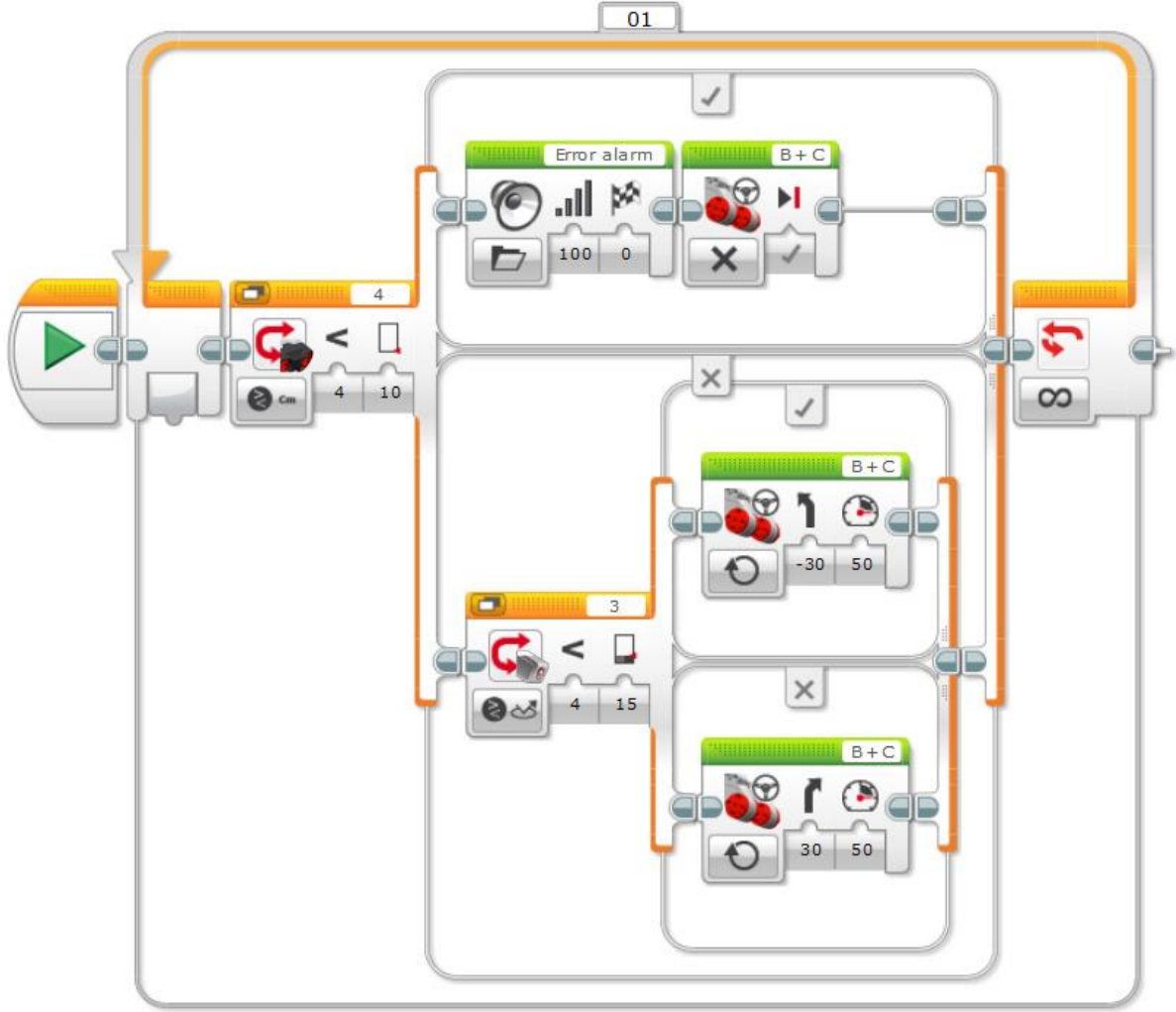
Fikir üretme: Bu aşamada öğrenciler yukarıda belirlenen işlemleri robotun nasıl gerçekleştirebileceği ile ilgili fikir yürütürler. Öğrenciler aşağıda belirtilen çözüm önerilerine benzer fikirler sunabilir.

- Renk sensörü ile mesafe sensörü eş zamanlı olarak kullanılmazdır.
- Renk sensörü zemine, mesafe sensörü ise karşıya bakacak şekilde kullanılmalıdır.
- İç içe switch blokları kullanılabilir.
- Öncelikli olarak engel olup olmadığı kontrol edilebilir.

2.2. Üret: Çizgi İzleyen ve Engelde Duran Robot

Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve robotlarla çalışarak istenilen görevi yerine getirirler.

Örnek program aşağıdaki resimde görülmektedir.



Resim 106. Örnek Program

2.3. Tasarla: Çizgi İzleyen ve Engeli Aşan Robot

Bu aşamada, bir önceki görevde pasif olarak engelin kaldırılmasını bekleyen robot yerine çizgi üzerinde engel ile karşılaştığında engeli aşabilecek bir robot programlanması istenir. Engel olarak oyuncak yapı blokları ile hazırlanmış küp kullanılabilir. Robot engel ile karşılaşınca engeli alıp kendi etrafında 180 derece döner ve engeli çizgi üzerine bırakır, sonra tekrar 180 derece dönerek engeli arkasında bırakıp yoluna devam eder. Gerekli kol tasarımını öğrenciler kendi kendilerine geliştirmelidir.

2.4. Üret: Çizgi İzleyen ve Engeli Aşan Robot

Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenilen görevi yerine getirirler.

3. DEĞERLENDİR

Gün sonunda öğrencilerle halka oluşturulur. Sensör isimleri kâğıtlara yazılır ve tombala oyunu misali öğrencilerden bir kâğıt seçmeleri istenir:

- Öğrencilerden seçilen sensörlerin özelliklerini anlatmaları ve bu sensörlerin hangi projelerde, nasıl kullanılabileceğini açıklamaları istenir.
- Sensörler ikili olarak gruplara dağıtılır. Öğrencilerden bu sensörlerin ikili olarak hangi projelerde, nasıl kullanılabileceğini açıklamaları istenir.
- Bütün sensörlerin bir arada kullanılabileceği projeler tartışılır.

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşınca kadar devam ettirilir.

4. İLAVE ETKİNLİK

4.1. Nesnelere Arasından Belirli Renklerdeki Toplamak

Bu etkinlikte matın "A" yüzeyindeki daire içerisinde iken kendi etrafında dönerek matın üzerinde bulunan nesnelere tarayacak bir robot tasarlanması ve programlanması gerekir. Robotun taradığı nesnelere arasından sadece kırmızı renkte olanları (farklı bir renk de olabilir) daire içerisine getirmesi istenir. Robotun bunu yaparken farklı renkteki nesnelere getirmemesi de sağlanmalıdır. Bu etkinlikte mesafe, renk ve açı sensörleri birlikte kullanılabilir.

Proje Hazırlıyorum

Bu hafta itibarıyla proje çalışmalarına başlanması önerilmektedir. Projenin "hazırlık" süreci için planlamalar yapılmalıdır. Detaylar için EK'ler incelenebilir.

9. Hafta: Değişkenler

Ön bilgi:

- Öğrenciler robot kavramını bilir.
- Öğrenciler robot setiyle farklı robot tasarımları yapmıştır.
- Öğrenciler robot setini programlamak için grafik arayüzünü kullanmıştır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler tuğlanın çeşitli sesleri çıkarması ve tuğla ekran görüntüsünün düzenlenmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler dokunma, mesafe, açı ve renk sensörlerini farklı amaçlar için programlama adımlarını oluşturmuştur.

Haftanın Kazanımları:

- Öğrenciler değişken tanımını yapar.
- Öğrenciler değişken türlerini açıklar.
- Öğrenciler değişkenlerle sensörleri bir arada kullanır.
- Öğrenciler değişkenleri kullanarak program akışını değiştirir.
- Öğrenciler değişkenleri problem çözümünde kullanır.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin değişken kavramını öğrenerek değişken tanımlamasıdır. Sonraki süreçte, öğrencilerin çeşitli problemlerin çözümünde değişkenleri tanımlamak ve kullanmak için gerekli düzenlemeleri yapabilmesine yardımcı olmaktır.

Kullanılacak Malzemeler:

Robot seti, bilgisayar, mesafe sensörü, renk sensörü, mat (çalışma alanı).

Haftanın İşlenişi:

Gözle: Değişkenlerin tanımlanması ve kullanılmasını çeşitli örnekler üzerinde inceleme.

Uygula: Değişkenlerin tanımlanmasını ve kullanılmasını çeşitli örnekler üzerinde uygulama.

Tasarla: Robotun karşısına çıkan engelin sağından geçme, çizgi takip ederken önüne çıkan engeli aşip tekrar çizgiyi takip etme, renkleri sayma ve yerde renklerle oluşturulan şifreli sayıyı bulma işlemlerini değişkenler kullanarak gerçekleştirebilmesi için gerekli tasarımları yapma.

Üret: Verilen görevleri programlama.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği.

1. GÖZLE VE UYGULA

1.1. Gözle: Değişkenler

Programcılar yazdıkları programlar çalıştırıldığında genellikle bilgisayardan bazı değerleri hatırlamasını isterler. Bilgisayar bu değerleri “RAM” olarak adlandırılan bellekte saklar. Bilgisayarın “RAM” belleğinde sakladığı bu değerlere programcılar değişken adını verir. Değişkenler aracılığı ile bilgisayarın belleğinde çeşitli değerler saklanabilir, değişkenin içerisinde saklanan değerler daha sonra istenildiğinde çeşitli işlemler için kullanılabilir ve istenilirse değiştirilebilir. Öğrencilerin bir sınavdan aldığı puanlar, boy uzunlukları ve bir sınıftaki öğrenci sayısı değişkenlere örnektir. Programcılar değişkenleri çeşitli isimlerle adlandırır. Örneğin, 5A sınıfındaki öğrencilerin sayısının bir değişken olarak saklanması istenirse bu değişkene “5aOgrenciSayisi” adı verilebilir. Değişken ismi olarak “5aÖğrenciSayısı” yerine “5aOgrenciSayisi” yazılmasının nedeni, “*Lego robotlarını programlamak için kullanılan EV3 yazılımının Ö, Ş, Ğ, Ü, İ, Ç ve ı gibi Türkçe karakterlere*” izin vermemesidir. Ayrıca, *değişken isimlerinde kullanılması yasak olan karakterler (örneğin “!” karakteri) girildiğinde EV3 yazılımı uyarı verir.* Programcı değişken isimlerini bu kurallara göre dilediği gibi belirleyebilir. Fakat değişken isimleri uzun olduğunda onları okumak zorlaşabilir. Bu yüzden değişken isimlerinin kısa olması programcının işini kolaylaştırır. Örneğin, 5A sınıfındaki öğrenci sayısının tutulacağı değişkenin ismi değişkenin işlevini anımsatacak şekilde kısaca “5aOgrSay” olarak da belirlenebilir.

1.2. Uygula: Grup Sorusu

Bu sorular her bir gruba sorulur ve yanıtlar sınıfta tartışılır:

- Bilgisayar oyunlarında her oyuncunun bir puanı vardır. Bilgisayar oyunlarında oyuncunun puanı değişkene örnek olabilir mi? Yanıtınızı sebepleriyle açıklayınız.
- Günlük yaşamınızdan iki farklı değişken örneği bulunuz. Bu değişkenlere istediğiniz bir isim veriniz. Fakat verdiğiniz ismin kısa ve anlamlı olmasına dikkat ediniz.

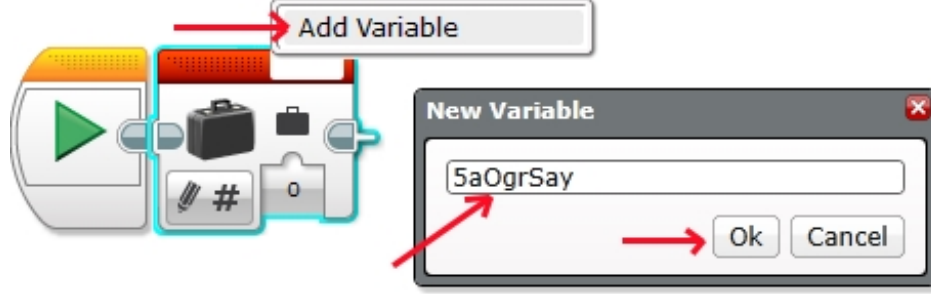
1.3. Gözle: EV3 Yazılımında Değişken Tanımlama

Değişken tanımlamak için Veri İşlemleri (Data Operations) sekmesindeki “Değişken (Variable)” bloğu kullanılır. Değişkeni oluşturup ona bir isim vermek için “Değişken (Variable)” bloğunun sağ üst köşesindeki boşluğa fare imleci ile tıklanır.



Resim 107. Adım 1

Aşağıdaki resimde de görüldüğü gibi ortaya çıkan menüden “Değişken Ekle (Add Variable)” seçilir ve açılan metin kutusunun içerisine değişkenin ismi yazılır. Son olarak “Tamam (Ok)” butonuna tıklanır.



Resim 108. Adım 2

EV3 yazılımında “metinsel” (text), “mantıksal” (logic) ve “sayısal” (numeric) değerler değişkenler içerisinde saklanabilir. Metinsel değişkenlerde saklanacak değer bir metindir. Örneğin, “Merhaba Dünya” ifadesi bir metinsel değişkendir. Mantıksal değişkenlerde ise “doğru” (true) veya “yanlış” (false) değerleri saklanabilir. Sayısal değişkenlerin içerisinde “5” ve “2.5” gibi sayısal değerler saklanır. Çeşidi ne olursa olsun bir değişkene öncelikle değer ataması yapılır. Bunun için “Değişken (Variable)” bloğunda “Yaz (Write)” modu seçilir ve çıkan menüden değişkenin tipi belirlenir. Aşağıdaki resimde görünen örnekte “5aOgrSay” değişkenine “36” değeri verildiği görülebilir.

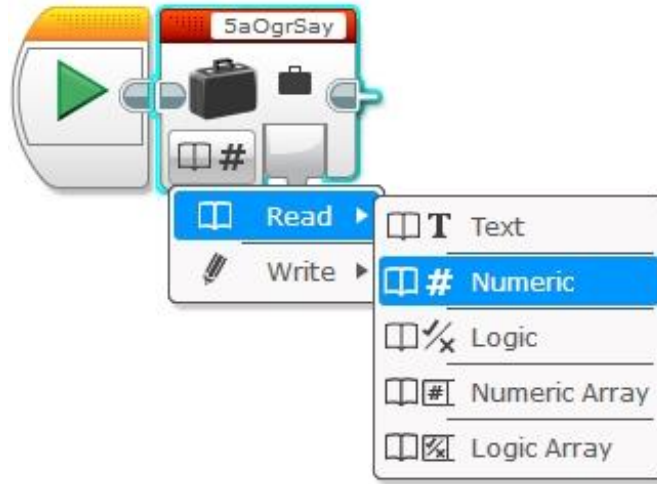


Resim 109. Adım 3

Artık bu değer programın içerisinde “5aOgrSay” ismi ile saklanır. İstenildiğinde farklı işlemler için bu değişken kullanılabilir veya bu değişkenin değeri değiştirilebilir. Değişkenin değerini değiştirmek için “Yaz (Write)” modunda değişkene yeni bir değer vermek yeterlidir (**Not:** Burada “36” değeri “45” ile değiştirilerek değişken için nasıl yeni bir değer atanabileceği öğrencilere gösterilir).

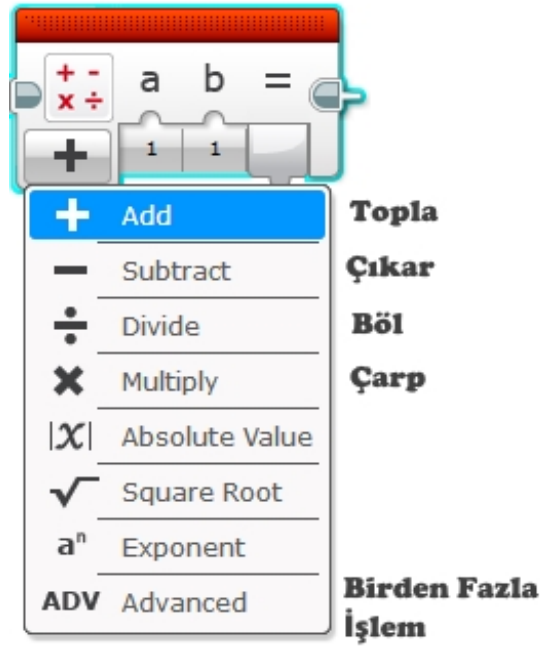
1.4. Gözle ve Uygula: EV3 Yazılımında Değişken Kullanma

Sayısal değişkenler ile yapılabilecek işlemlere örnek olarak “5A” sınıfındaki öğrenci sayısı ile “5B” sınıfındaki öğrenci sayısı toplanmak istenmiş olsun. “5A” sınıfında “36”, “5B” sınıfında ise “28” öğrenci vardır. Öncelikle bu değişkenler oluşturulup bunlara sırasıyla “36” ve “28” değerleri atanmalıdır. Ardından bu değerler “Okuma (Read)” modunda toplama işlemine aktarılmalıdır. Aşağıdaki resimde okuma moduna nasıl gelineceği gösterilmektedir.



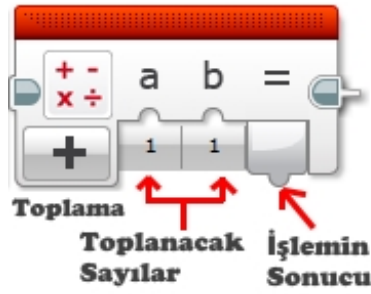
Resim 110. Okuma Modu

Toplama işlemi yapmak için resimde gösterildiği gibi Veri İşlemleri (Data Operations) sekmesindeki “matematik (math)” bloğu kullanılmalıdır. “Math” bloğu ile toplama, çıkarma, çarpma ve bölme gibi işlemler yapılabilir.



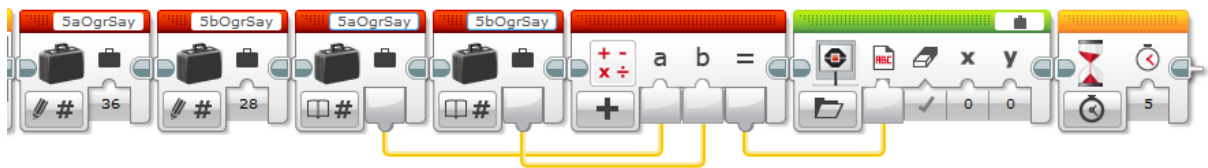
Resim 111. Matematiksel İşlemler

Öncelikle math bloğunda yapılacak işlem seçilir. Toplama işlemi yapılacağı için aşağıdaki resimde toplama işlemi seçildiği görülmektedir. Bu blokta işleme girecek sayılar için “a” ve “b” başlıklarında iki girdi kutucuğu ve sonuç için “=” başlığı altında çıktı kutucuğu bulunur. “a” ve “b” kutucuklarına girilen değerler toplanır ve sonuç (=) kutucuğuna aktarılır.



Resim 112. Toplama İşlemi

Aşağıdaki resimde “5A” ve “5B” sınıfındaki öğrenci sayılarını toplayıp robotun ekranına yazan program bulunuyor. Rehber öğretmen bu programı öğrencilere uygulamalı bir şekilde anlatmalıdır.



Resim 113. Örnek Program

1.5. Gözle ve Uygula: Karşısına Engel Çıktığında Sağından Geçen Robot

Bu etkinlikte karşısına engel çıktığında engelin sağından geçen robot tasarlanır. Etkinlik için bir mesafe sensörü gerekir.

Dikkat

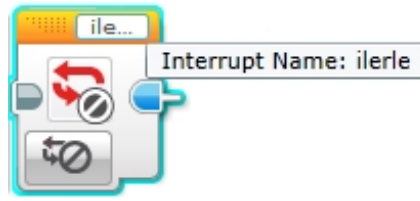
Rehber öğretmen bu kodun çalıştırılmasını öncelikle öğrencilere göstermelidir. Robotun çalışması için geniş bir alana ihtiyaç vardır. Bu etkinlikte dersliğin zemini kullanılabilir. Robotun karşısına çıkacak engel için herhangi bir nesne kullanılabilir veya rehber öğretmen robot ilerlerken ayağını robotun önüne engel olarak koyabilir. Robot ilerlerken birkaç defa robotun rotasına engel eklenebilir. Süreç esnasında rehber öğretmen robotların ortamdaki herhangi bir nesneye çarpmamasına da dikkat etmelidir.

Bu programda amaç “robotun bir çizgiyi takip etmesi” değildir. *Bu programda amaç, zeminde düz bir hatta ilerleyen robotun karşısına 35 cm’den daha yakın bir engel çıktığında engelin sağından geçerek yeniden ileriye doğru hareket etmesini sağlamaktır.* Engeli aştıktan sonraki yön ile engelden önceki yön bire bir aynı olmak zorunda değildir. Robotun engel öncesi ve engel sonrası benzer yönlerde ilerlemesi yeterlidir. Problemin çözümü için aşağıdaki algoritma kullanılabilir.

- i. Bir değişken tanımlanmalı ve bu değişkenin değeri “0” olmalıdır.
- ii. Robot bir engel ile karşılaşana kadar ileriye doğru gitmelidir.
- iii. Eğer robot bir engele 35 cm’den fazla yaklaşırsa değişkenin değeri “1” olmalıdır.
- iv. Değişkenin değeri “1” ise (a) robot durmalı, (b) “90” derece sağa dönmeli, (c) tekerlerini bir tur döndürüp tekrar durmalı, (d) “90” derece sola dönmeli, (e) değişkenin değerini “0” yapmalı ve (f) sabit hızla düz doğrultuda devam etmelidir.
- v. iii ve iv numaralı adımlar tekrarlanmalıdır.

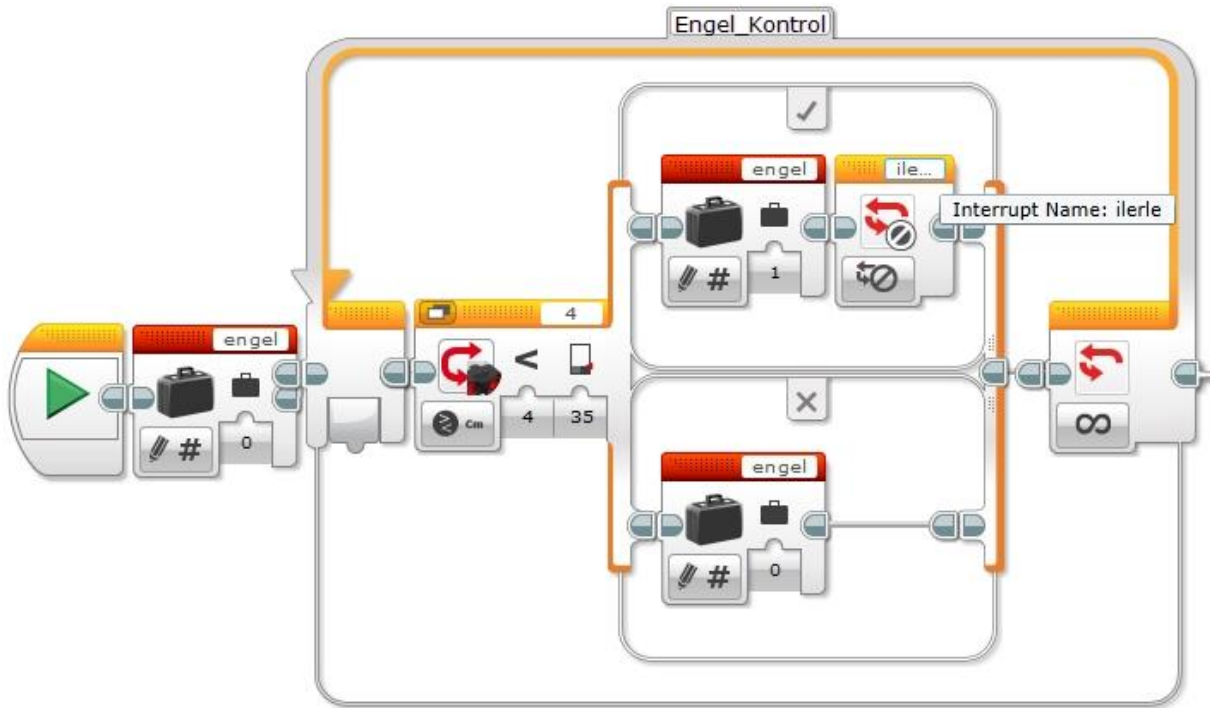
Bu program paralel çalışan iki kod hâlinde tasarlanır. Paralel kodlardan bir tanesi robotun önüne engel çıkıp çıkmadığını kontrol eder. İkinci kod ise robotun önünde engel yoksa düz ilerlemesini engel olduğunda ise engeli aşmasını sağlar.

Bu programın oluşturulması için döngünün çalışmasının nasıl sonlandırılacağı bilinmelidir. Birçok programlama dilinde “break” komutu kullanılarak döngü sonlandırılır. EV3 yazılımında ise Akış Kontrolü (Flow Control) sekmesinde bulunan “döngü kesme (loop interrupt)” bloğu ile döngü sonlandırılır. Aşağıdaki resimde “döngü kesme (loop interrupt)” bloğunun ilerle adlı döngünün çalışmasını sonlandırmak için nasıl kullanıldığı görülmektedir.



Resim 114. Interrupt Bloğu

Öncelikle paralel kodlardan ilki olan, engel olup olmadığını kontrol eden kod yazılabilir. Bu kod için engel isimli bir değişken oluşturulur. Robotun karşısında 35 cm'den yakın bir engel varsa engel değişkeninin değeri "1", engel yok ise engel değişkeninin değeri "0" olarak atanır. Engel ile karşılaşıldığında ilerle komutunun durdurulması gerekir. Bunun için engel değişkeninin 1'e eşit olması durumunda "döngü kesme (loop interrupt)" komutunun çalıştırılması gerekir. Aşağıdaki resimde bu işlemler için düzenlenen kod blokları görülmektedir.

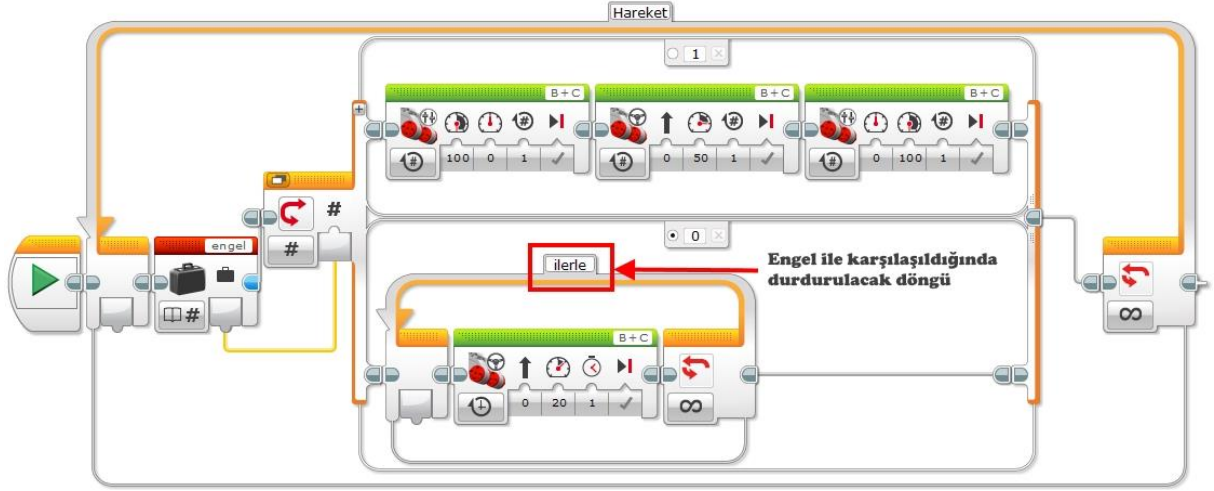


Resim 115. Birinci Kod

Paralel kodlardan ikincisinin görevi ise robotun aşağıdaki resimde gösterildiği şekilde sıralı işlemleri yapabilmesini sağlamaktır.

- i. Eğer karşıda engel yoksa (engel=0) düz bir şekilde ilerlemek,
- ii. Eğer karşıda engel varsa (engel=1) sağa dönüp bir teker turu ilerlemek ve
- iii. Sola dönmek.

Aşağıdaki resimde bu işlemler için düzenlenen kod blokları görülmektedir.



Resim 116. İkinci Kod

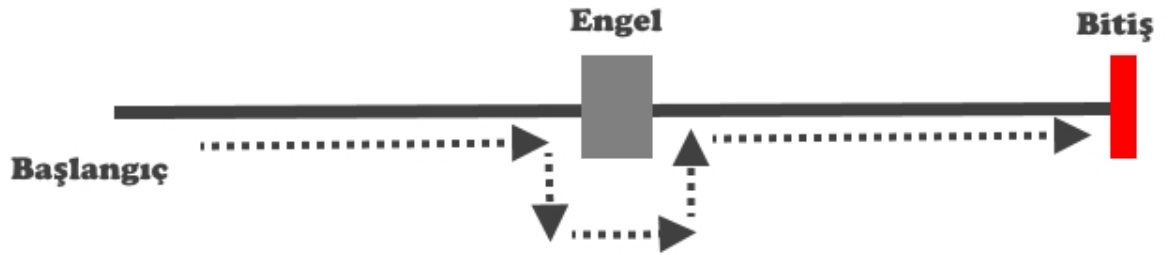
2. TASARLA VE ÜRET

2.1. Tasarla: Çizgi Takip Ederken Engel Aşan Robot

Rehber öğretmen çizgi takip ederken engel aşan robotun programının yazılmasını öğrencilerden ister. Bu programda robotun aşağıdaki sıralı işlemleri yapması sağlanır.

- Herhangi bir engelle karşılaşmadığında çizgiyi takip etmek.
- Engel ile karşılaştığında çizgiden ayrılıp engelin sağından geçmek.
- Engeli geçtikten sonra yeniden çizgi üzerine gelip çizgi üzerinden ilerlemeye devam etmek
- Kırmızı renkteki bitiş çizgisine geldiğinde durmak.

Robotun yapması gerekli görevin şeması aşağıdaki resimde görülmektedir . Görev için birden fazla engel kullanılabilir.



Resim 117. Görev Şeması

Programı oluşturmaya başlamadan önce grupların tasarlama adımı için yukarıda bir örneği verilen tanımlama ve fikir üretme süreçlerini gerçekleştirilmesi gerekir.

2.2. Üret: Çizgi Takip Ederken Engel Aşan Robot

Öğrencilerden tanımladıkları ve hakkında fikir ürettikleri programı tamamlamaları ve kontrol edip eksikliklerini gidermeleri istenir.

3. DEĞERLENDİR

Bu aşamada hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşüncelerini sağlamaktır. Bu sayede öğrenciler; problem çözme yetenekleri, dersin konusu ve kendileri ile ilgili gözlemler yaparak öğrendikleri yeni konuları ve kendilerini değerlendirmekle beraber sonraki çalışmalarını planlamak için de fırsat elde edeceklerdir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- En çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Yeteri kadar tartışma ortamı oluşmazsa rehber öğretmen aşağıdaki soruları kullanarak tartışma ortamı yaratmaya çalışır.
 - Çizgi takip ederken engel aşan ve tekrar çizgi takip etmeye devam eden robot uygulamasında, engeli aştıktan sonra robotun çizgiyi tekrar takip etmesi için ne kullandınız? Başka hangi yöntemlerle bu görevi yerine getirebilirdiniz?
 - Renk sayacı uygulamasında, sadece farklı renklerde (renk geçişlerinde) robotun sayacı artırmasını nasıl sağladınız? (İlave aktivite derste yapılabilirse)
 - Şifreli sayıyı bulma yarışmasında, değişkenleri kullanmadan beyaz renklerin sayılmamasını sağlayabilir miydiniz? (İlave aktivite derste yapılabilirse)
- Problemi çözerken ne gibi sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla anlaşmazlığa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne/neler öğrendiniz?

Değerlendirme, öğrencileri sıkmadan, her bir soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilir.

5. İLAVE ETKİNLİK

5.1. Tasarla: Şifreli Sayıyı Bulma Yarışması

Bu sorunun öğrenciler tarafından çözülebilmesi için öncelikle öğrencilerin ikilik sayı tabanını ve sabitleri bilmesi gerekir. Sabitlerin öğretmen tarafından öğrencilere anlatılacağı varsayıldığından burada bahsedilmeyecektir. Rehber öğretmen sabitleri kendisi anlatmalıdır. İkilik sayı tabanı şu şekilde anlatılabilir.

- i. Öğrencilerden aşağıdaki örüntünün kuralını bulmaları istenir.
1 2 4 8 16 32 64 ...
- ii. Örüntünün devamında hangi sayıların geleceği sorulur.
- iii. Öğrencilere ikilik sayı tabanındaki sayıların 0 ve 1 rakamından oluştuğu anlatılır. Birkaç tane sayı örneği verilir.

- iv. Öğrencilere ikilik bir sayının basamaklarına ayrılması aşağıdaki kutu tekniği ile anlatılır:

1011 sayısı aşağıdaki şekilde basamaklarına ayrılır.

1	0	1	1
---	---	---	---

- v. Yukarıda bulunan örüntünün elemanları sağdan sola doğru ve basamaklarına ayrılan sayı ise soldan sağa doğru olacak şekilde alt alta yazılır. Her bir sütundaki iki sayı çarpılır ve tüm sonuçlar toplanır.

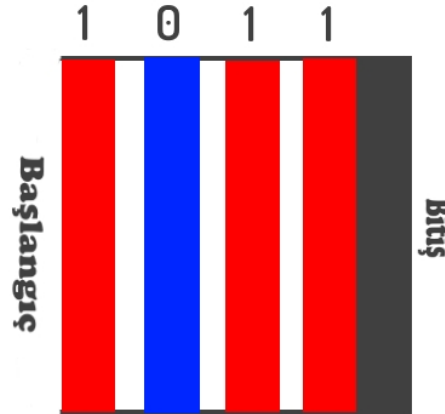
8	4	2	1
1	0	1	1

$8*1+4*0+2*1+1*1=11$. Sonuç ikilik tabandaki bir sayının (1011) onluk tabandaki karşılığıdır. Bu şekilde ikilik tabandaki sayılar onluk tabana çevrilir.

- vi. Öğrencilere bir iki tane örnek verilerek onluk tabana çevirmeleri istenir.

Görev Tanımı:

Renkler kullanılarak şifreli sayılar oluşturulur. Örneğin aşağıdaki resimde görüldüğü üzere kırmızı renk 1'e mavi renk ise 0'a karşılık gelir. Beyaz rengin sayısal bir karşılığı yoktur. Siyah renk ise şifrenin bittiğini gösterir. Kırmızı ve mavi renklere karşılık gelen sayılar sırasıyla yan yana yazılarak şifreli sayı bulunur. Aşağıdaki resimde şifreli sayı ikilik tabandaki "1011" sayıdır.



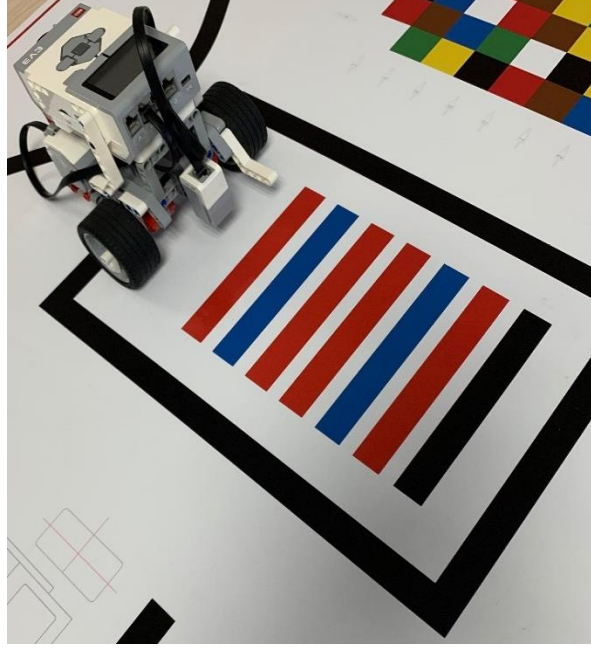
Resim 118. Şeklin İkilik Taban Karşılığı

Yukarıdaki resimde görüldüğü üzere mat üzerinde şifreli bir sayı bulunmaktadır. Robot matta soldan sağa doğru ilerlemeli ve siyah kısma geldiğinde durarak şifreyi ekranına yazmalıdır. Programı doğru çalışan ve şifreli sayının onluk tabandaki karşılığını ilk bulan grup yarışmayı kazanır.

Programı oluşturmaya başlamadan önce grupların tasarlama adımı için tanımlama ve fikir üretme süreçlerini gerçekleştirmesi gerekir.

Dikkat

Öğrenciler robotu çok hızlı hareket ettirmemelidirler. Hızlı giden robot yanlışlıkla masadan düşebilir. Öğrenciler robotun hızı konusunda uyarılmalıdır.



Resim 119. Kullanılacak Mat Bölümü

5.2. Üret: Şifreli Sayı Bulma Yarışması

Öğrencilerden tanımladıkları ve hakkında fikir ürettikleri programı tamamlamaları ve kontrol edip eksikliklerini gidermeleri istenir.

5.3. Tasarla: Renk Sayacı

Bu programın amacı renk sensörüne gösterilen farklı renklerin sayısını ekrana yazdırmaktır. Program renk sensörüne gösterilen her yeni renk için sayaç değerini bir artırır ve bu değeri ekrana yazar. Örneğin, program ilk çalıştığında ekrana “0” yazar. Ardından sensöre kırmızı renkli bir cisim gösterildiğinde ekrandaki değer “1”, siyah renkli bir cisim gösterildiğinde “2”, mavi renkli bir cisim gösterildiğinde “3” olur fakat yeniden mavi renkli bir cisim gösterilirse ekrandaki değerde bir artış olmaz. Gösterilen her yeni renk için ekrandaki değer “1” artar. Program dokunma sensörüne basılmasıyla çalışmayı durdurur.

Programı oluşturmaya başlamadan önce grupların tasarlama adımı için daha önce bir örneği verilen tanımlama ve fikir üretme süreçlerini gerçekleştirmesi gerekir.

Dikkat

Bu program için renk sensörüne gösterilmek üzere farklı renklerde cisimlere ihtiyaç vardır. (Bunun yerine mat üzerindeki renkler de kullanılabilir.). Renk sensörüne gösterilecek cisimler ışığı fazla ışık yansıtırlarsa programın çalışmasını etkileyebilir. Bu yüzden mat cisimler tercih edilmelidir.

5.4. Üret: Renk Sayacı

Öğrencilerden tanımladıkları ve hakkında fikir ürettikleri programı tamamlamaları ve kontrol edip eksikliklerini gidermeleri istenir.

5. 5. Modifiye Renk Sayacı

Bir önceki renk sayacı etkinliğinde robot kendine gösterilen cisimlerin renklerini algılayıp bunları sayıyordu. Bu programın bir önceki programdan tek farkı robotun yerde bulunan ardışık renkler üzerinde ilerlerken kaç adet yeni renk bulunduğunu saymasıdır. Rehber öğretmen zamanı olan öğrencilerden bu programı yapmalarını isteyebilir.

Proje Hazırlıyorum

Bu hafta itibarıyla, proje çalışmalarına başlanması önerilmektedir. Projenin “empati” süreci için planlamalar yapılmalıdır. Detaylar için EK’ler incelenebilir.

10. Hafta: Paralel İşlemler

Ön Bilgi:

- Öğrenciler robot kavramını temel düzeyde bilir.
- Öğrenciler robot setiyle farklı robot tasarımları yapmıştır.
- Öğrenciler robotu programlamak için EV3 yazılımını kullanmıştır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler farklı sensörlerin kullanıldığı programlar oluşturmuştur.

Haftanın Kazanımları:

- Öğrenciler paralel işlemleri uygular.
- Öğrenciler EV3 yazılımının deney arayüzünü kullanır.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin robotların herhangi bir veya birden fazla sensörünün algıladığı verileri bir dosya olarak kaydetmelerini ve bu verilerin zamana göre değişim grafiklerini inceleyip yorumlamalarını sağlamaktır. Ayrıca birbiriyle çelişmemek kaydıyla programların paralel olarak çalıştırılması, yani robotun farklı görevleri eş zamanlı olarak yerine getirecek şekilde programlanabilmesi amaçlanmaktadır.

Kullanılacak Malzemeler:

Robot seti, bilgisayar, mat (çalışma alanı).

Haftanın İşlenişi:

Gözle: Paralel işlemleri gerçekleştirmek için birden fazla başla bloğu kullanma, kablo ile blokları birbirlerine bağlama ve deney arayüzünde sensör verilerinin grafiklerini inceleme.

Uygula: Paralel işlemlerin gerçekleştirilmesi ve deney arayüzünde verileri inceleme.

Tasarla: Otonom olarak veri toplayan robot oluşturma ve verileri değerlendirme.

Üret: İstenen programları gerçekleştirme.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği.

1. GÖZLE VE UYGULA

1.1. Gözle: Paralel İşlemler

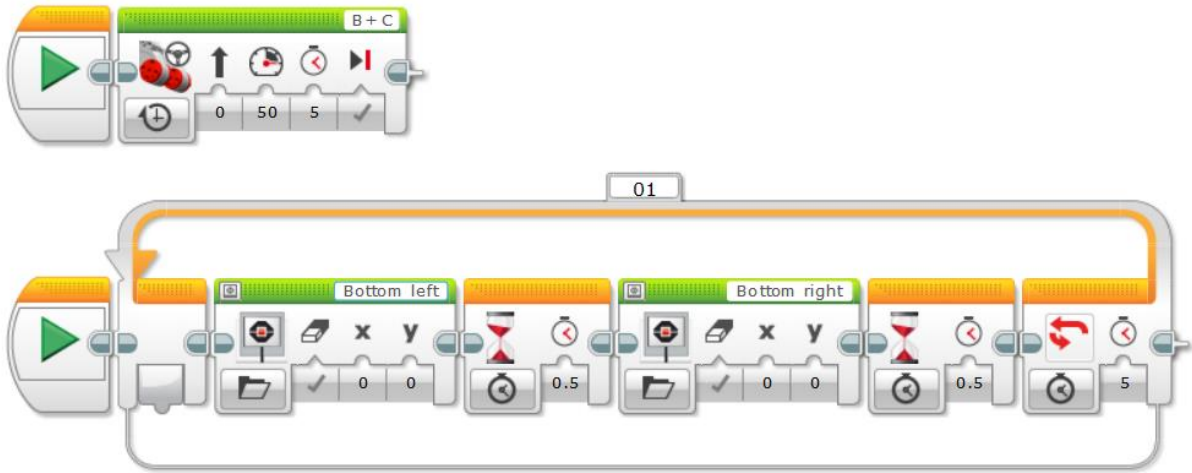
Bu haftada öğrencilere birden fazla program bloğunun eş zamanlı olarak nasıl kullanılacağı öğretilir. Bugüne kadar yaptıkları çalışmalarda program genel olarak bir kanal üzerindeki belirtilen komutları sırayla gerçekleştiriyordu. Bu haftada ise iki işlemin, örneğin robotun ileriye doğru hareket etmesi ile orta motor aracılığıyla kolun hareket etmesinin eş zamanlı olarak nasıl gerçekleştirilebileceği konusu işlenir.

Robotun iki işlemi aynı anda yapması iki şekilde gerçekleştirilebilir.

Birden Fazla Başla Bloğu Kullanmak

EV3 yazılımında programlama alanında bir adet başla bloğu bulunur; diğer programlama blokları başla bloğuna eklenerek bir sıra halinde program oluşturulur. Akış Kontrolü (Flow Kontrol) sekmesinde yer alan “Başla (Start)” bloğu programlama alanına sürüklenerek birden fazla program akış kanalı oluşturulabilir. Böylece program iki farklı kanalı eş zamanlı olarak çalıştırır.

Öğrencilere 5 saniye boyunca, ileri giderken sağa ve sola bakan göz ifadelerinin sürekli değiştiği (yani etrafına bakınıyormuş gibi ilerleyen) robot programı gösterilir.



Resim 120. Örnek Program

Hazırlanan programda iki adet “başla (start)” bloğu bulunur. Program başladığı anda her iki başla bloğu da çalışmaya başlar. Böylece program tek bir kanaldan değil, iki kanaldan eş zamanlı olarak ilerler. Üst kanalda B ve C motorları ile ileriye doğru 5 saniyelik hareket devam ederken alt kanalda iki farklı göz resmi 5 saniye boyunca 0,5'er saniyelik aralıklarla sürekli değişir.

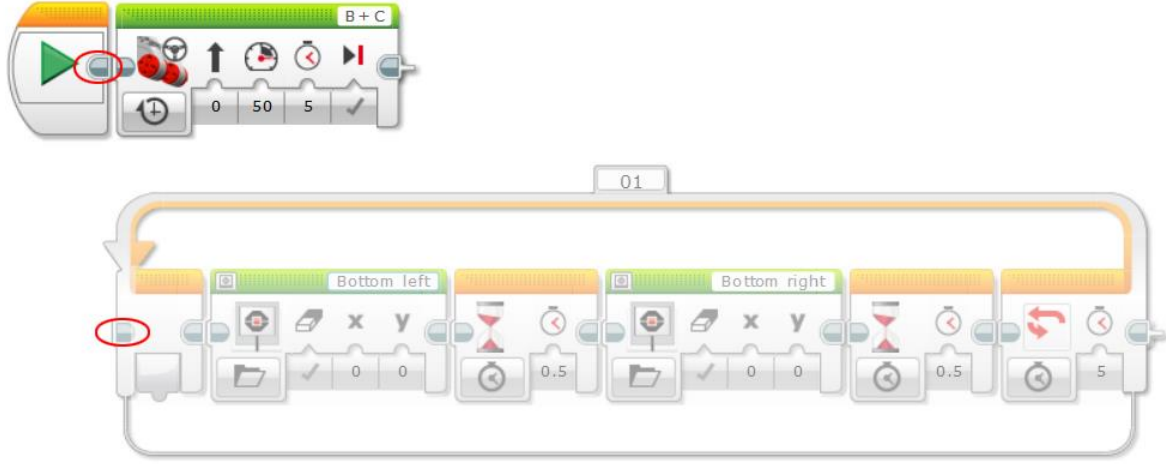
Not

Paralel işlemlerde önemli olan nokta, kanallarda birbiriyle çelişen komutların yer almamasıdır. Örneğin, üst kanalda B ve C motorları ileri yönde hareketi sağlarken alt kanalda B ve C motorlarına geriye doğru hareket komutu verilmemelidir. Bu durumda robotun hangi kanaldaki komutu gerçekleştireceğini tahmin etmek olanaksızdır.

Blokları Kablo ile Bağlamak

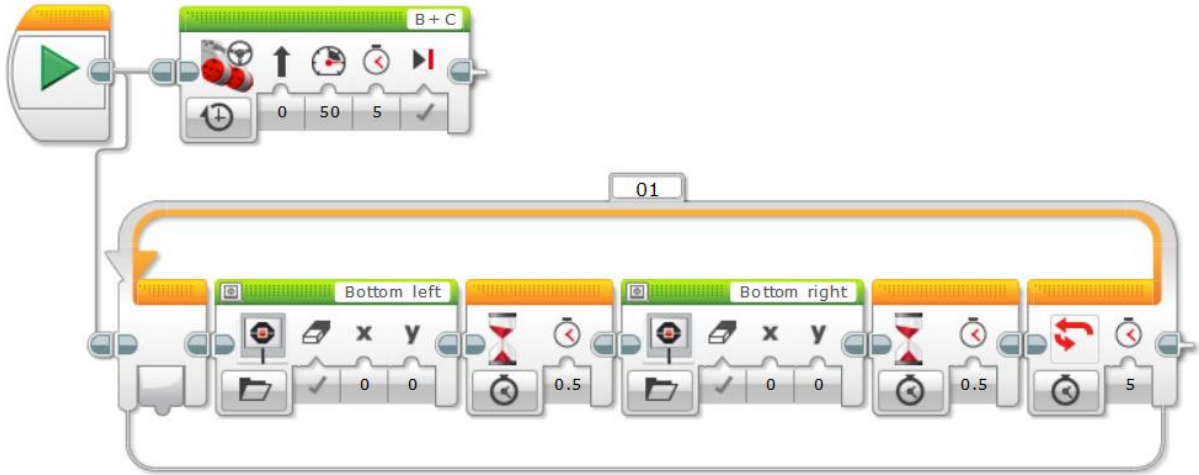
Birden fazla başla bloğu kullanıldığında, programda gerçekleştirilmesi istenen paralel işlemlerin mutlaka program başladığı andan itibaren başlaması gerekir. Robotun belirli işlemleri bir kanalda gerçekleştirdikten sonra, paralel işlemleri gerçekleştirmesi istendiğinde kablo bağlantısı kullanmak gerekir.

Önceki örnekte iki farklı başla bloğu ile gerçekleştirilen robotun hareketi ve ekranda görüntünün değiştirilmesi etkinliği kablo ile de gerçekleştirilebilir.



Resim 121. Adım 1

Kablo bağlantısı yapılmak istenen program bloğunun sonunda yer alan bağlantı bölgesi üzerine fare ile sol tıklanarak bağlanmak istenen program bloğunun başlangıcında yer alan bağlantı bölgesine sürüklenir. Böylece aktif olmayan program blokları aktif hâle getirilir.



Resim 122. Adım 2

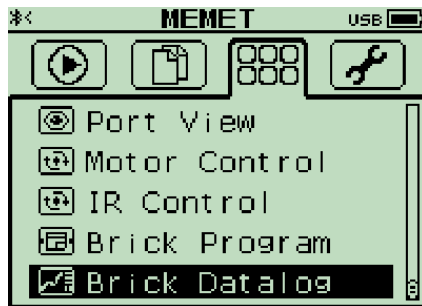
Bu yöntemle, birden fazla başla komutu kullanmadan, programın istenen noktasından itibaren paralel işlemler başlatılabilir. Yani programın ilerleyen herhangi bir noktasında robota eş zamanlı görevler verilebilir.

1.2. Uygula: Paralel İşlemler

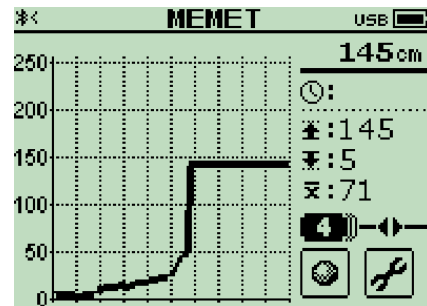
Öğrencilerden hareket hâlindeyken orta motora bağlı kolu indirerek herhangi bir nesneyi alan robot programını oluşturmaları istenir. Programda paralel işlemleri kullanmaları gerektiği, robotun nesneyi alırken durmaması ve hareketine devam etmesi gerektiği vurgulanır.

1.3. Gözle: Deney Arayüzü ve Veri Kaydı

EV3 robot setini kullanarak farklı yöntemlerle “Veri Kaydı (Data Logging)” yapılabilir. Bunlardan birincisi, akıllı tuğla üzerinde üçüncü sekmeden “Tuğla Veri Kaydı (Brick Datalog)” menüsünün seçilmesidir (Robotun Firmware versiyonuna bağlı olarak Brick Datalog menüsü olmayabilir).



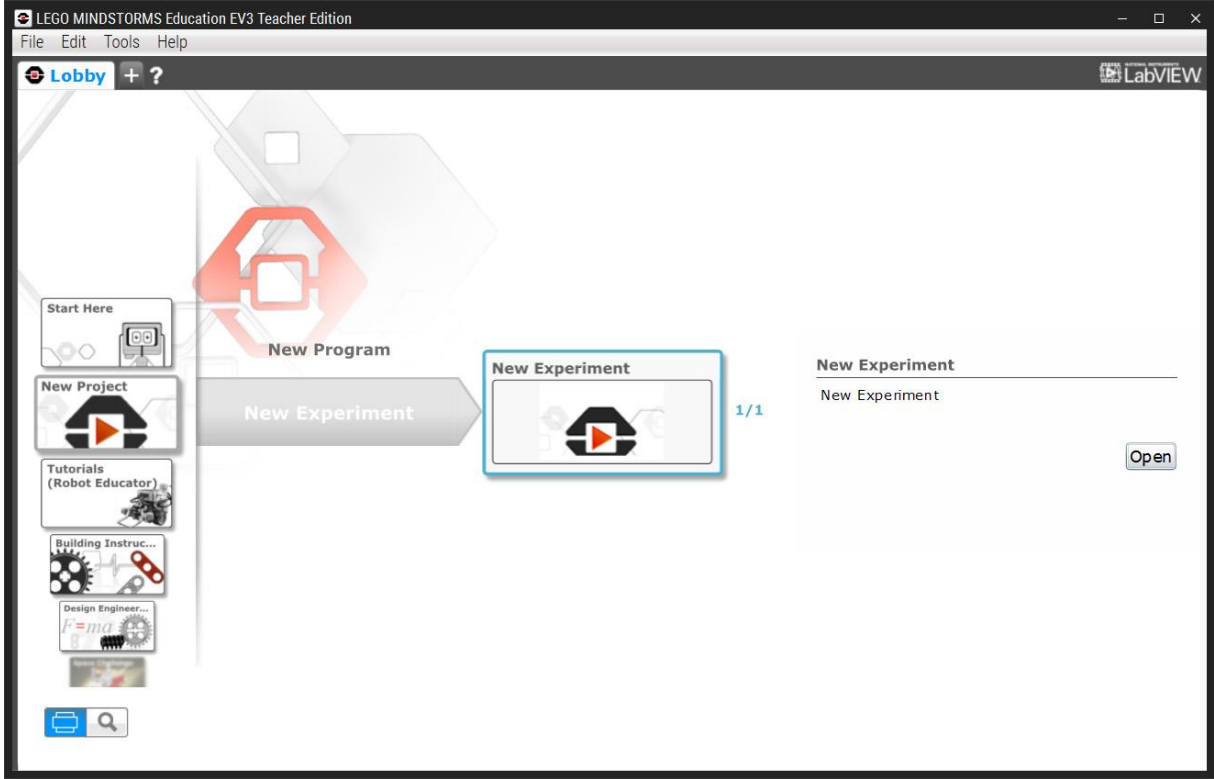
Resim 123. Veri Kaydı Seçimi



Resim 124. Veri Grafiği

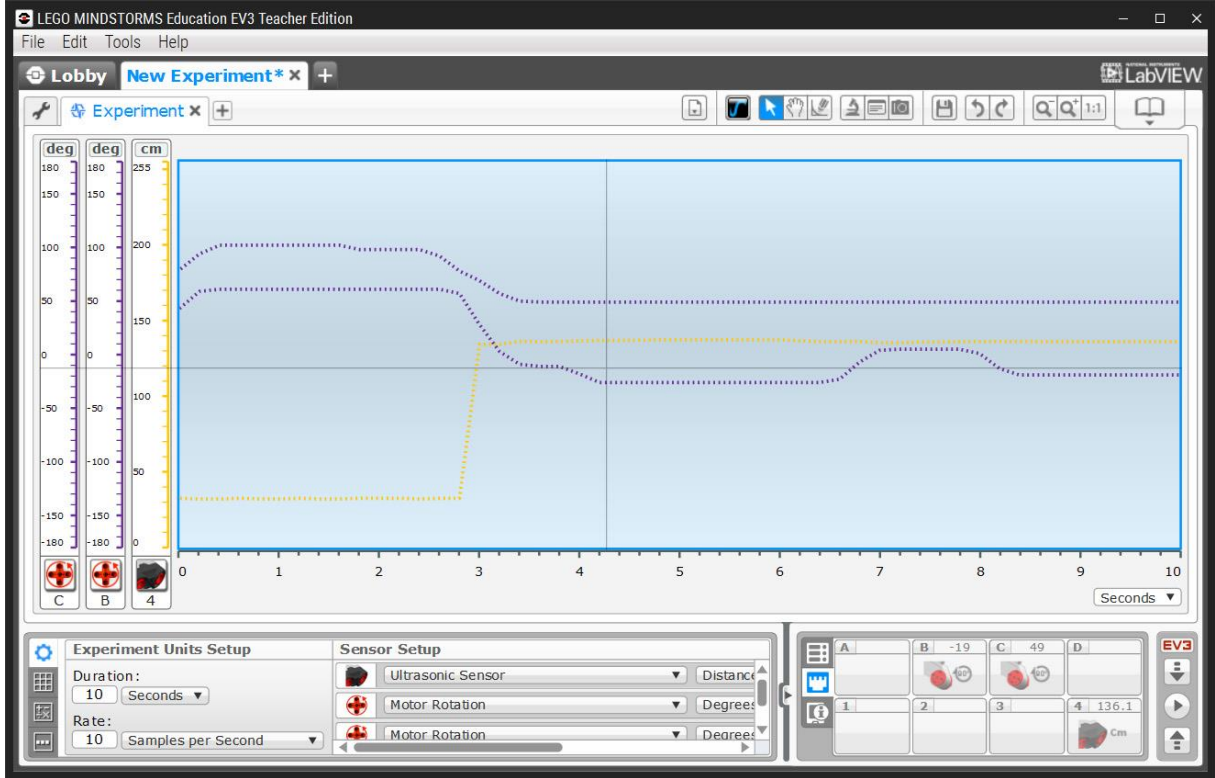
İstlenen sensör veya motor portu seçilerek o porttan gelen veri grafiğinin ekranda çizildiği gözlemlenebilir.

İkinci olarak EV3 yazılımı robotun sensörleri aracılığı ile topladığı verilerin grafiğini çizdirerek üzerinde çalışmaya olanak veren deney arayüzüne sahiptir. Deney arayüzü, Lobby ekranında **New Project > New Experiment > Open** yolu izlenerek açılabilir.



Resim 125. Deney Arayüzü

Deney arayüzü açıldığında, eğer robot bilgisayara USB veya bluetooth ile bağlı ise deney ekranında eş zamanlı olarak robotun sensörlerinden alınan verilerin grafiğinin çizildiği görülecektir.



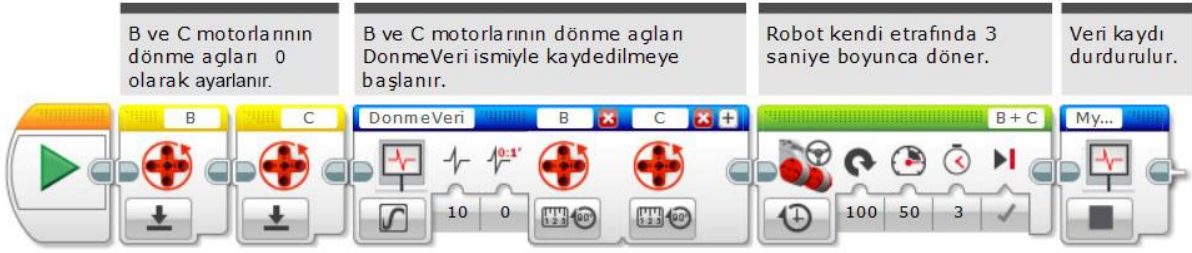
Resim 126. İlgili Veri Grafiği

Öncelikli olarak robotun sadece B ve C motorları verilerini göstermesini sağlayınız, bu amaçla “Sensör Kurulumu (Sensor Setup)” bölümünden diğer sensör verilerini X işaretini tıklayarak kapatınız. Robotun hareketsizken ve bir motorun elle yavaşça döndürülmesiyle oluşan grafikleri inceleyiniz ve öğrencilerle tartışınız. Öğrencilerin motorun dönme açısını ve zaman grafiğini anlamasını sağlayınız. Ayrıca benzer bir etkinliği motor verilerini kapatıp sadece mesafe sensörü verisini açarak yapınız. Robotun bir engelle yaklaşırken ve engelden uzaklaşırken oluşan grafiği öğrencilerle tartışınız.

Son olarak robotun bağımsız olarak istenen sensör verilerini akıllı tuğla içinde bir dosyaya kaydetmesinin sağlanması ve bu dosyanın daha sonra deney arayüzünde incelenmek üzere açılabilmesi de mümkündür. Bu amaçla öncelikle robotun istenen sensör verilerini toplamasını ve bu verileri bir dosyaya kaydetmesini sağlayan programın hazırlanması gerekir. EV3 yazılımı programlama arayüzünde, mavi renkli “İleri Düzey (Advanced)” sekmesinde yer alan “Veri Kaydı (Data Logging)” bloğu ile istenen sensör verilerinin kaydedilmesi sağlanır.

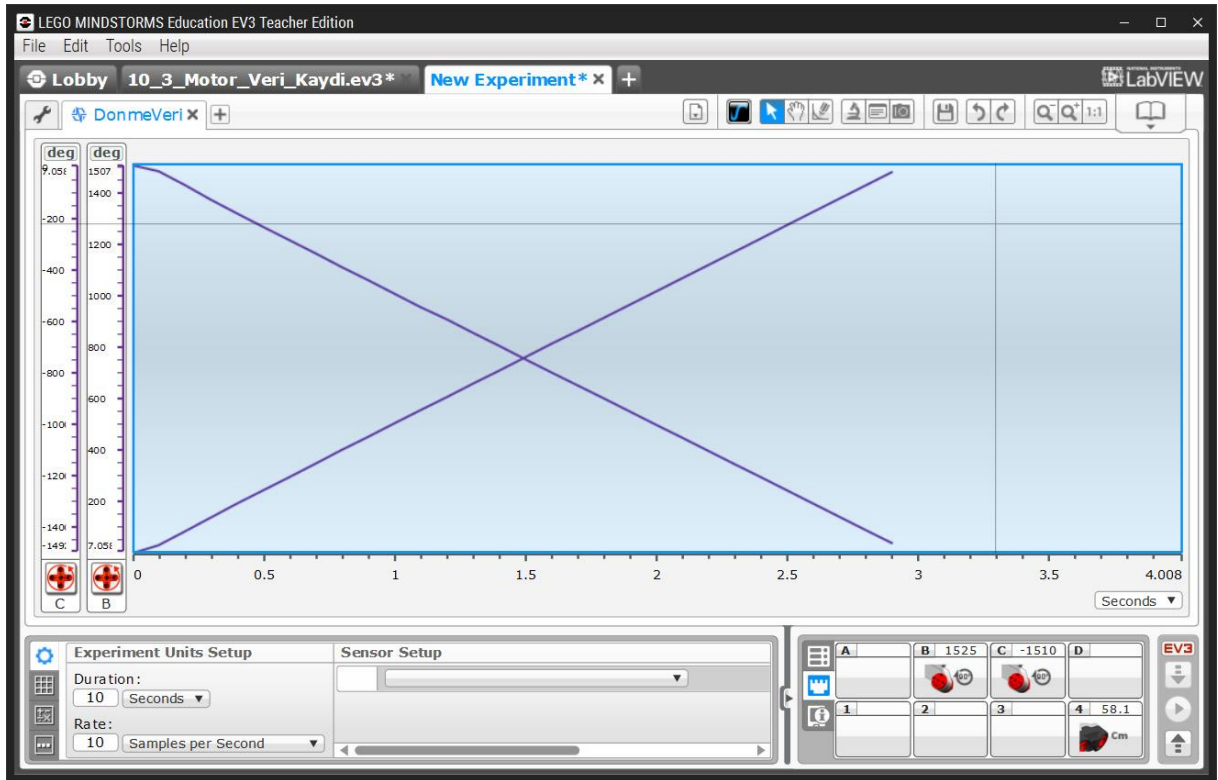
1.4. Uygula: Motor Hareket Verisinin İncelenmesi

Öğrencilerden kendi etrafında dönen robotun B ve C motorlarının dönme verilerini kaydedecek programı hazırlamaları ve programı çalıştırarak verileri robotun akıllı tuğlasına kaydetmeleri istenir. Sonrasında EV3 yazılımı deney arayüzünde verilerin grafiğini inceleyip tartışmaları istenir. Bu amaçla aşağıdaki gibi bir program ile motor verilerinin tuğla içinde DonmeVeri dosyasına kaydedilmesi sağlanır.



Resim 127. Örnek Program

Öğrencilerden, robotu bilgisayara bağlayarak deney arayüzünde, “Araçlar (Tools)” menüsünden, “Veri Kaydı Dosya Yöneticisi (Data Log File Manager)” açılarak çalıştırılan program klasörüne kaydedilmiş DonmeVeri dosyasını açmaları ve tartışmaları istenir. Aşağıda DonmeVeri dosyası deney arayüzünde açıldığında oluşan grafik görülmektedir.



Resim 128. Veri Grafiği

1.5. Uygula: Motor ve Mesafe Sensörü Verilerinin Karşılaştırılması

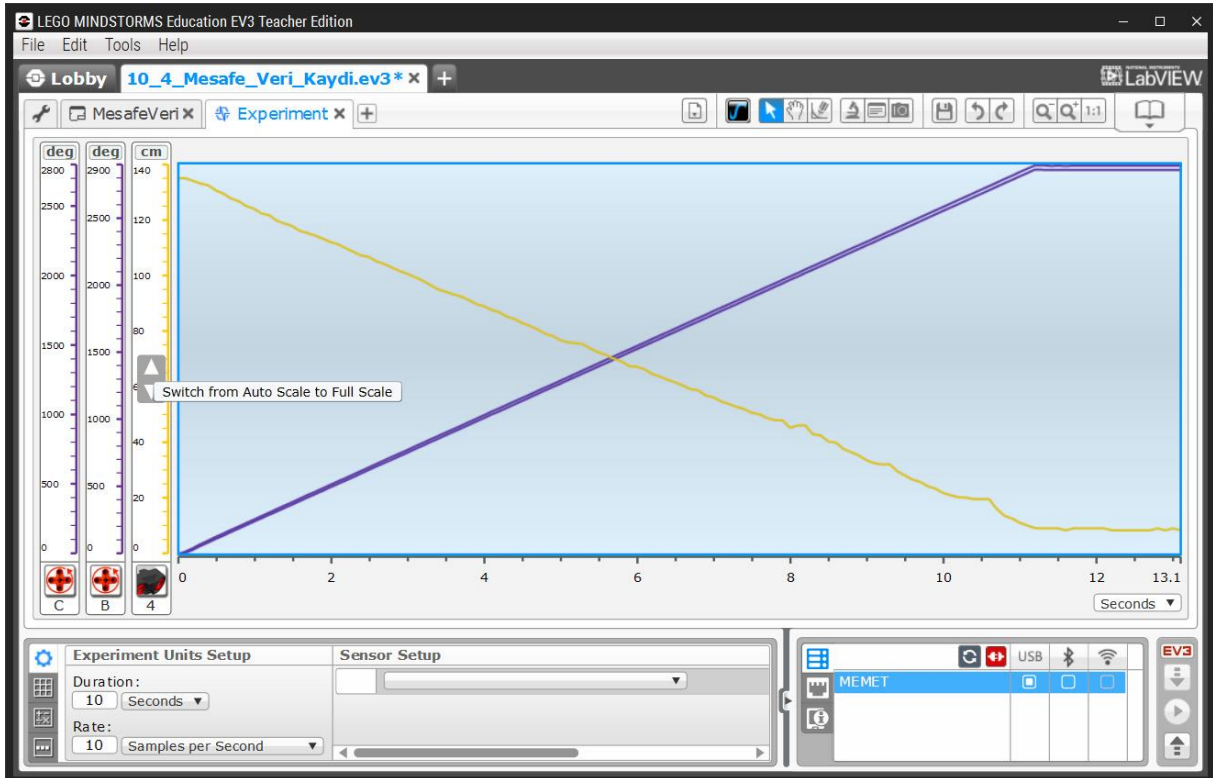
Bu etkinlikteki amaç, robot bir engele yaklaşırken mesafe sensörü ve motorlardan alınan verilerin deney arayüzünde incelenmesidir. Öğrencilerden robotlarını bir engele 10 cm kalıncaya kadar düz gidecek şekilde programlamaları istenir. Robotun bu hareketi esnasında motorlarının dönme açısını ve mesafe sensörünün ölçtüğü mesafeyi veri dosyası olarak kaydetmesi gerektiği belirtilir. Verinin daha rahat incelenebilmesi için öğrencilerden motorların gücünü 10-25 olarak ayarlamaları istenir.

Deney arayüzünde bu verilerin grafiğini çizdirerek tartışmaları istenir. Örnek program aşağıda sunulmuştur.



Resim 129. Örnek Program

UltraVeri dosyası EV3 deney arayüzünde açıldığında aşağıdaki gibi bir grafik elde edilir. Bu grafik öğrencilerle tartışılır.



Resim 130. Veri Grafiği

Not

Robot bilgisayara bağlı olduğu sürece robota takılı olan sensörlerin canlı verileri de deney arayüzünde görülür. Daha sade bir arayüzde çalışmak için alt kısımda bulunan

Sensör Kurulumu (Sensor Setup) bölümünde canlı olarak gösterilen sensörler kapatılabilir. Ayrıca grafiği çizilen sensör verilerini ekrana sığdırabilmek için fare sensör sütunu üzerine tıklanarak otomatik sığdırma (auto scale) veya tam sığdırma (full scale) görünümleri arasında geçiş yapılabilir.

2. TASARLA

2.1. Tasarla: Renk Tahmini Oyunu

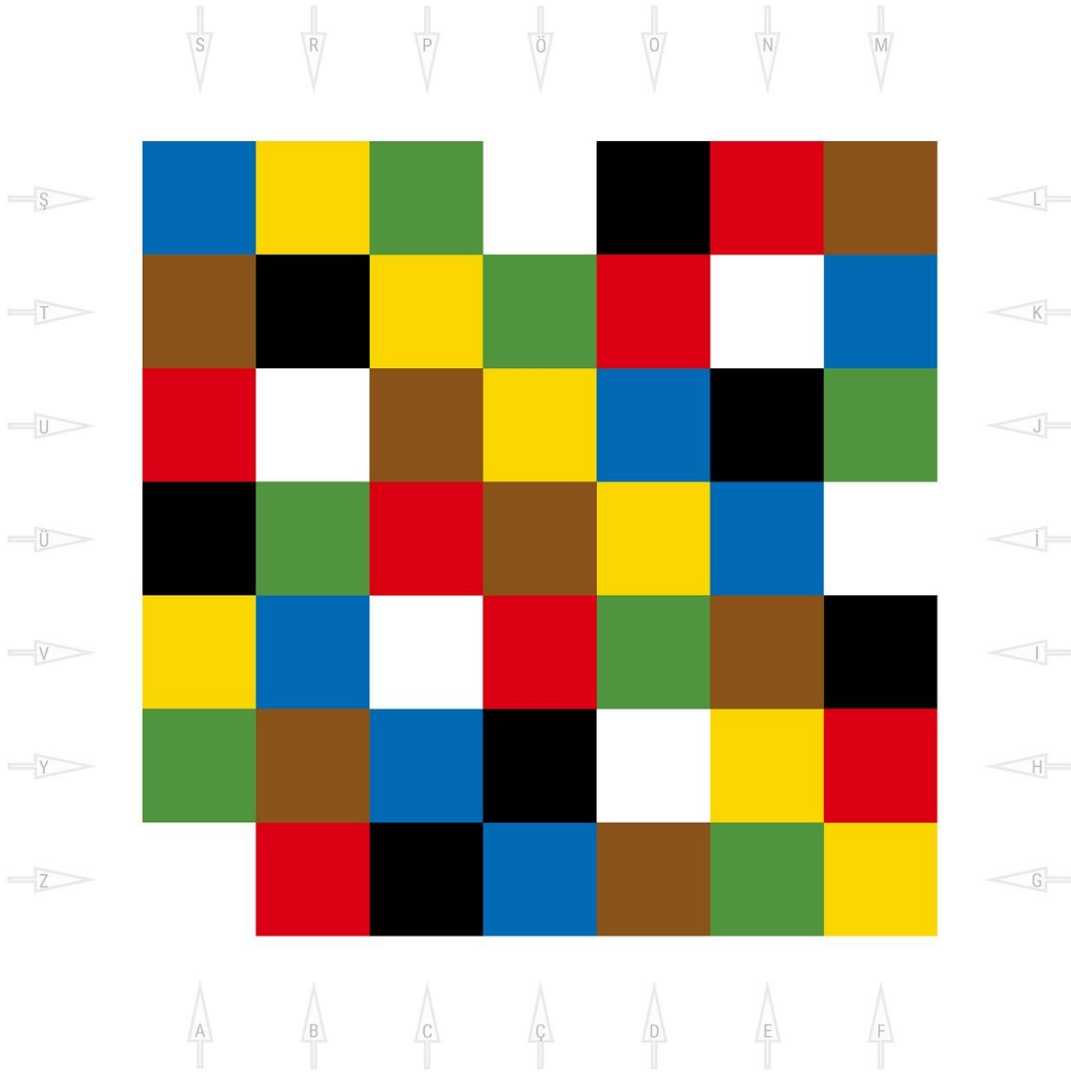
Bu etkinlikte öğrencilere, robotlarını veri toplama sürecinde görmeyecekleri, robotlarını bilinmeyen bir noktaya veri toplamak amacıyla gönderecekleri ve toplanan verileri analiz ederek robotlarının hangi renkte bir zeminde gittiğini tahmin etmelerinin isteneceği söylenir.

Not

Renk sensörlü temel tasarıma (colour sensor down - driving base) erişmek için gerekli yönergeye aşağıdaki yollarla ulaşılabilir:

- i. <https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-rem-color-sensor-down-driving-base-d30ed30610c3d6647d56e17bc64cf6e2.pdf>
- ii. EV3 yazılımı > Lobby > Building Instructions > Building Ideas > Color Sensor Down - Driving Base
- iii. Robot seti ile birlikte gelen kitapçığa bakılabilir.

Gruptaki öğrencilerden biri, robotunu 23 cm düz gidecek ve ilerlerken de zeminden yansıyan ışık miktarına ait verileri toplayacak şekilde programlar. Grup arkadaşı, programlanan robotu mat üzerinde yer alan renkler damasındaki herhangi bir rotada (A'dan Z'ye isimlendirilmiştir) düz gidecek şekilde çalıştırarak veri kaydının yapılmasını sağlar. Robotu çalıştıran öğrenci renkler damasının 1 cm dışındaki beyaz alandan ölçümü başlatmalı ve robot beyaz alana çıkınca ölçüm sonlanmalıdır (renkler damasının bir kenarı 21 cm uzunluğundadır). Veri toplama sürecinde programı hazırlayan öğrenci robotun hangi rotadan ilerlediğini görmemelidir. Sonraki aşamada, veri dosyası kaydedilen robot, programı hazırlayan öğrenciye verilir ve öğrenci veri dosyasını deney arayüzünde açıp analiz ederek robotun hangi rotada ilerlediğini tahmin eder. Yansıyan ışık miktarından rengi tahmin etmek güç olabilir, öğrenciler beyaz ve siyah alanlardan çıkarımda bulunabilir ve takıldıkları noktalarda robot üzerinde üçüncü sekmede bulunan Giriş Görünümü (Port View) menüsünden mat üzerindeki renklerin yansıma miktarını ölçebilirler. Öğrenciler rolleri değişerek etkinliği tekrar ederler.



Resim 131. İlgili Mat Bölgesi

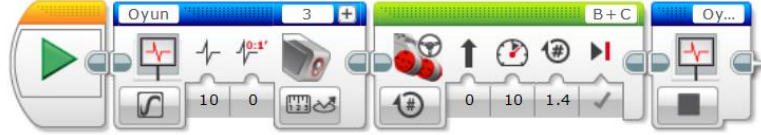
Tanımlama: Öğrencilerin problemi tanımlamaları gerekir. Öğrenciler, problemi çözmelerine yardımcı olacak şu soruların cevaplarını kendi aralarında tartışılar:

- Robot zeminden yansıyan ışık miktarını nasıl algılayıp kaydedebilir?
- Yansıyan ışık miktarı renkleri ayırt etmek amacıyla kullanılabilir mi?
- Işığın en fazla yansıyan renk hangisidir? Işığın en az yansıyan renk hangisidir?
- Grafik nasıl yorumlanır?

Fikir üretme: Bu aşamada öğrencilerden robotun yukarıdaki işlemleri nasıl gerçekleştirebileceği ile ilgili fikir yürütmeleri beklenir.

3. ÜRET

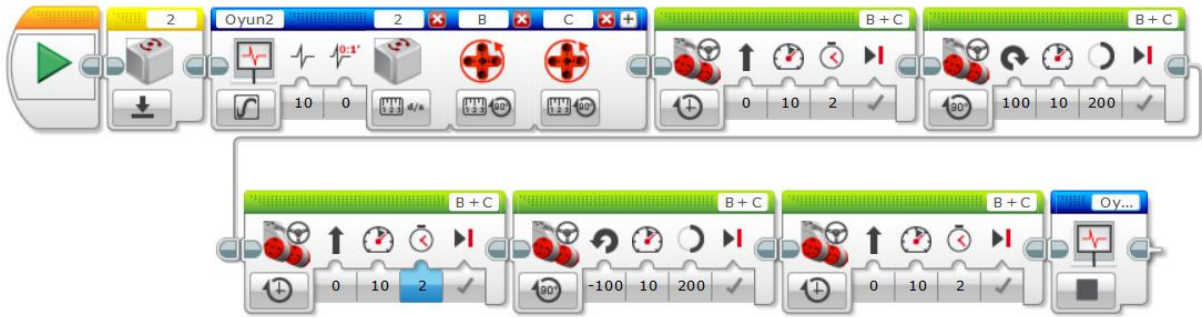
Birinci öğrenci gerekli programı hazırlayıp robota yükledikten sonra (öğrencinin robotun ilerlediği rotayı görmemesi sağlanarak), ikinci öğrenci renkler daması üzerinde robotu çalıştırıp veri toplama sürecini gerçekleştirir ve robotu birinci öğrenciye teslim eder. Birinci öğrenci verileri inceleyerek robotun geçtiği rotayı tahmin etmeye çalışır. Sonrasında öğrenciler rolleri değişirler. Örnek program aşağıda sunulmuştur.



Resim 132. Örnek Program

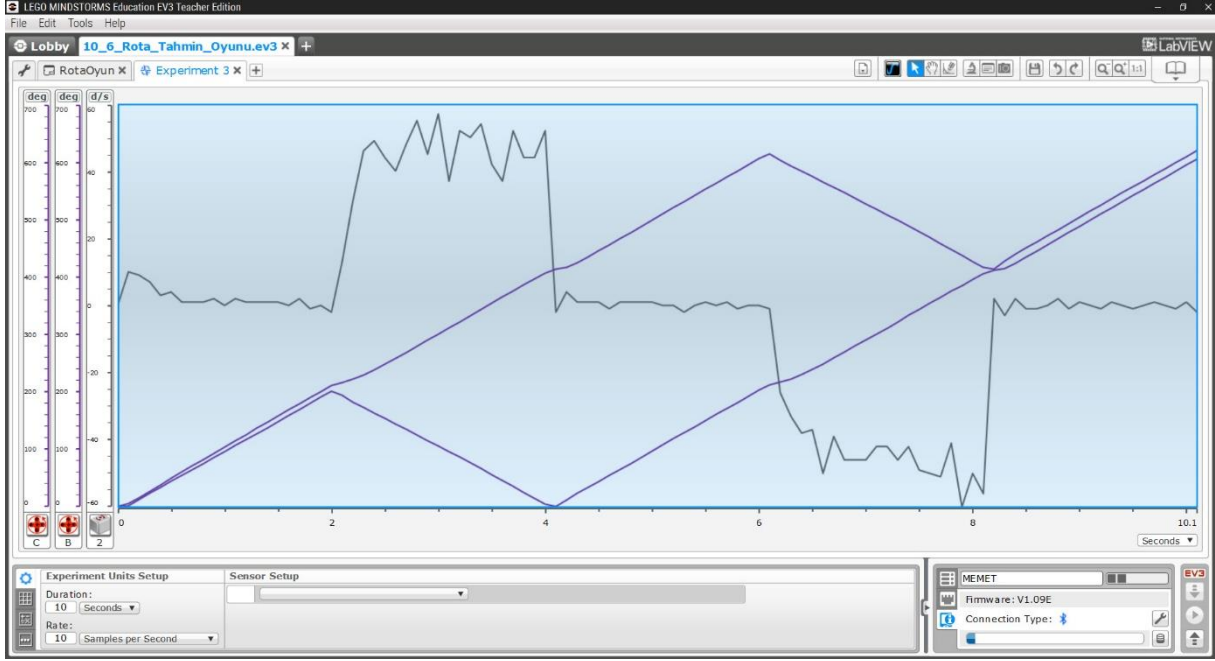
3.1. Motor ve Açık Sensör Verilerini Kullanarak Rota Oluşturma

Bu etkinlikte rehber öğretmen, öğrencilere açık sensörü, B motoru ve C motorundan kaydedilen veri setini hazır olarak verir ve öğrencilerden veri setini inceleyerek robotun izlediği yolu bir kâğıda çizmelerini ister.



Resim 133. Örnek Program

Rehber öğretmen, yukarıdaki programa benzer bir programı kendi bilgisayarında hazırlar. Öğrencilerin program bloklarını ve robotun çalışmasını görmelerine izin vermez. Program çalıştırılınca robot 2 saniye süreyle ileri gider, sonrasında 90 derece sağa döner. Tekrar 2 saniyelik düz bir hareketin ardından bu defa 90 derece sola döner ve 2 saniye daha düz gider. Bu esnada robotun B, C ve açı sensörü verileri Oyun2 adlı dosyaya kaydedilir. Rehber öğretmen tüm bu süreci öğrencilerin göremeyeceği şekilde gerçekleştirir. Sonrasında tüm gruplarla yalnızca Oyun2 veri dosyası paylaşılır. Öğrenciler veri dosyasını deney arayüzünde açıp veri setini inceleyerek robotun izlediği yolu kâğıda çizerler. Deney arayüzünde açılan dosyada aşağıdaki gibi bir grafik oluşur.



Resim 134. Veri Grafiği

Not

Etkinlik daha fazla hareket bloğu ile çeşitlendirilebilir. Rehber öğretmen eğitim öncesinde veri dosyasını hazırlayabilir, etkinlik esnasında da öğrencilerden beklenenleri açıklayıp dosyayı paylaşabilir. Etkinlik öncesinde birden çok veri dosyası hazırlanmalıdır.

4. DEĞERLENDİR

Bu aşamada hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşüncelerini sağlamaktır. Bu sayede öğrenciler süreç içinde deneyimlediklerini ve öğrendiklerini gözden geçirerek kendilerini değerlendirme ve tanıma fırsatı elde edeceklerdir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Robotun birden fazla işlemi eş zamanlı gerçekleştirebildiği görevlere örnekler veriniz.
- Robot paralel işlemler yapabilmesi için nasıl programlanmalı ve nelere dikkat edilmelidir.
- Robotun verileri toplayıp uzmanlara gönderdiği durumlara günlük hayattan örnekler veriniz.
- EV3 yazılımının deney arayüzünde neler yapıldığını özetleyiniz, önemini tartışınız.
- Bilim insanlarının çalışmaları ile bugün yaptığınız etkinlikleri kıyaslayınız.
- Bugün neler öğrendiniz?
- Grup arkadaşınızdan ne/neler öğrendiniz?

Değerlendirme, öğrencileri sıkmadan, her soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilir.

5. İLAVE ETKİNLİK

5.1. Diğer Grubun Hareketini Tahmin Edip Uygulanan Programı Oluşturma

Bu etkinlik iki grubun karşılıklı etkileşimde olacağı şekilde yapılır. Gruplar aynı anda hareket ve dönme gerektiren bir program yazar, çalıştırır ve buradaki verileri (açı ve hareket) robota kaydeder. Daha sonra bu verileri diğer gruba paylaşır. Kaydı alan grup kaydı oluşturulan robotun hareketini tahmin eder ve aynı hareketleri gerçekleştirecek programı hazırlar.

Not

Bu etkinliğin amacına ulaşması için veri kaydı yapılırken ve veri kaydını oluşturan program yazılırken grupların birbirinin yaptığı işlemleri görmemesi gerekir.

Proje Hazırlıyorum

Bu haftada, proje çalışmalarına devam edilmesi önerilmektedir. Projenin “tanımlama” süreci için planlamalar yapılmalıdır. Detaylar için EK’ler incelenebilir.

11. Hafta: Kendi Bloğunu Oluşturma

Ön Bilgi:

- Öğrenciler robot setiyle farklı tasarımları yapmıştır.
- Öğrenciler robot setini programlamak için grafik arayüzünü kullanmıştır.
- Öğrenciler robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler farklı sensörleri birlikte veya paralel işlemlerde kullanmak için programlama adımlarını oluşturmuştur.
- Öğrenciler programlama adımlarında değişkenler kullanmayı, ileri seviye matematik ve mantık işlemleri kullanmayı deneyimlemiştir.

Haftanın Kazanımları:

- Öğrenciler robotun belirlenen işlemleri yapabilmesi için, verilen problemi daha küçük parçalara ayırabilir.
- Öğrenciler her bir küçük program parçası için yapılması gerekenleri tanımlayabilir.
- Öğrenciler her bir program parçasının çözümü için fonksiyonlar oluşturabilir.
- Öğrenciler küçük problemleri çözmek üzere geliştirilen fonksiyonların ana problemi çözmek için birlikte (senkronize) çalışması amacıyla gerekli programlama adımlarını oluşturabilir.
- Öğrenciler ortaya çıkan programın verimli ve etkili şekilde çalışması için gerekli önlemleri belirleyip uygulayabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin bir problemin çözümünde problemi küçük parçalara bölerek çözüm üretmelerini sağlamaktır. Öğrencilerin “Bloğum (My Block)” aracını kullanarak, belirli bir görevi yerine yetiren program parçacıklarını yeni bir blok olarak tanımlamalarını sağlamaktır.

Kullanılacak Malzemeler:

Robot seti, bilgisayar, sensörler ve mat (çalışma alanı).

Haftanın İşlenişi:

Gözle: Bir problemi daha küçük alt problemlere dönüştürme.

Uygula: Alt problemlerin program kodlarını oluşturma ve bu kodları fonksiyona dönüştürerek esas problemin çözüm sürecinde birlikte kullanma.

Tasarla: Belirlenen bir labirent için robot ve program tasarımı yapma.

Üret: İstenen robotu oluşturma ve oluşturulan robotun istenen işlemi yapması için gerekli programları oluşturma

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği

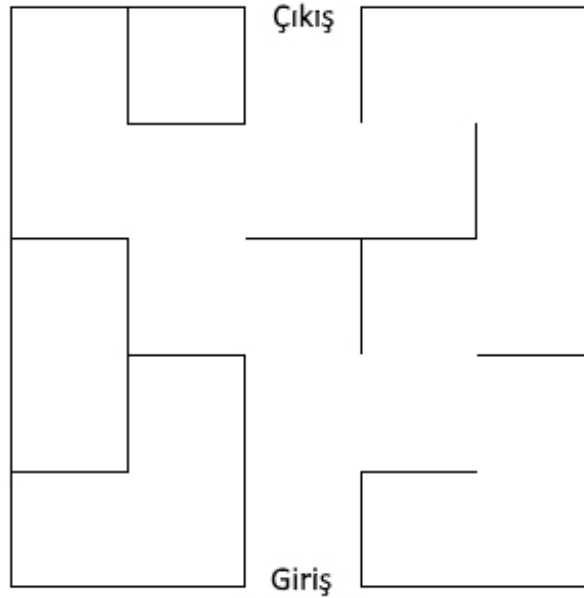
1. GÖZLE

1.1. Büyük Problemler Küçük Problemlerden Oluşabilir

Programcılar çoğu zaman büyük problemlerle karşı karşıya kalır. Böyle durumlarda, büyük bir problemi olduğu gibi çözmek yerine daha küçük parçalara ayırarak küçük problemler için çözüm üretmek ve çözümleri birleştirerek büyük problemin çözümüne ulaşmak iyi bir yaklaşım olabilir. Bu yöntem çeşitli avantajlar sağlayabilir. Bu avantajlardan en önemlisi zor bir problemin basitleştirilmesidir. Problemi bütün olarak çözmektense onu oluşturan her bir parçayla ayrı ayrı uğraşmak daha kolay olabilir. Bunun yanında, programcı büyük bir program oluşturmaya çabalamak yerine küçük programlarla uğraşacağından kod yazma ve hata ayıklama kolaylaşacaktır. Büyük bir programı takip etmek veya ondaki hatayı aramaktansa program parçacıklarının içindeki hatayı aramak daha kolay olacaktır. Ayrıca yazılan program parçaları, belirli bir işlem yapan birimler olarak düşünülebilir ve başka programların içinde yeniden kullanılabilir. Böylece bir programcı tarafından yazılan kod, kolaylıkla o programcı veya başka programcılar tarafından farklı projelerde yeniden kullanılabilir.

1.2. Labirent Çözme

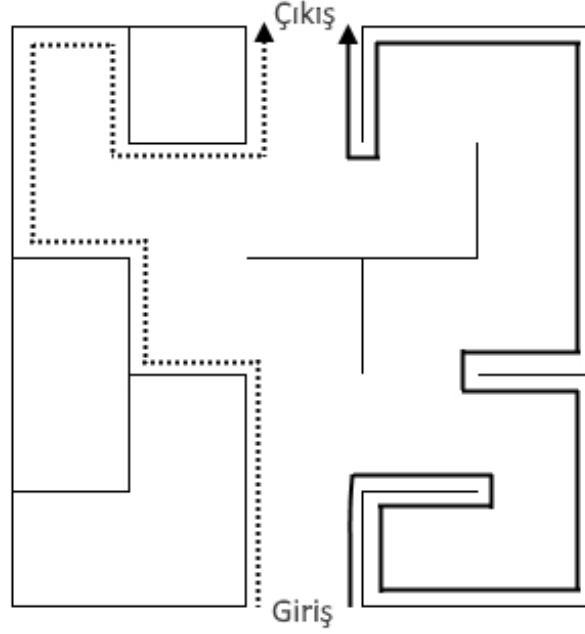
Rehber öğretmen dersin başlangıcında öğrencilere yapılacak etkinliklerde iki adet mesafe sensörü kullanılacağı için ikerli gruplar (4 öğrenci) hâlinde çalışacaklarını söyler. Aşağıda örnek bir labirent gösterilmiştir (**Dikkat:** Bu labirent öğretim amaçlıdır. Etkinlikte kullanılacak labirent 3.1. Labirent Yarışması başlığı altında sunulmuştur).



Resim 135. Örnek Labirent

Labirent problemlerinde esas olan, labirentten çıkmak için doğru yolu bulmaktır. Labirent problemlerinin çözümü için birçok algoritma bulunmaktadır. Sağ duvar algoritması en basitlerinden biridir. Girişten itibaren sürekli sağ duvar takip edilerek ilerlenirse sonunda çıkışa ulaşılabilir. Buna benzer şekilde girişten itibaren sol duvar takip edilerek de çıkışa ulaşılabilir. Aşağıda düz çizgi ile gösterilen çözüm sağ duvar

algoritmasının sonucu iken noktalı çizgi ile gösterilen çözüm sol duvar algoritmasının çözümüdür. Aslında bu iki algoritma aynıdır. Sadece takip edilecek duvarın yönü farklı seçilerek iki farklı çözüm oluşturulabilir.



Resim 136. Sağ/Sol Duvar Algoritması

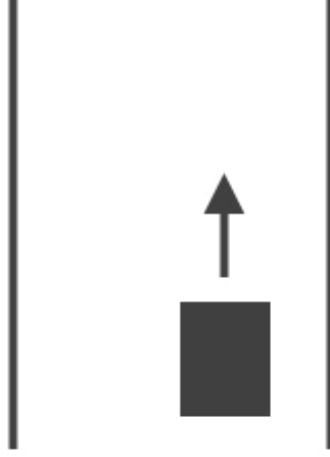
Sağ duvar algoritması her türlü labirent problemini çözmeyebilir. Eğer labirentin giriş veya çıkış noktası, labirentin kenarlarında değilse ya da labirentte hareket ederken başlanan yere geri dönme ihtimali varsa (döngü varsa) sağ duvar algoritması labirent problemini çözmede başarılı olmayabilir.

Sağ duvar algoritması çıkış için en kısa yolu bulamayabilir. Bu açıdan değerlendirildiğinde verimi düşük bir algoritmadır. Fakat bu dersin kapsamında sağ duvar algoritmasını göstermek uygun olacaktır.

1.3. Görevler

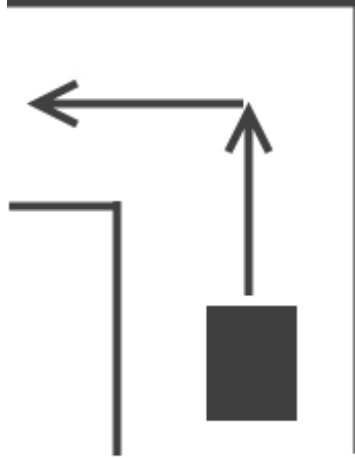
Sağ duvar algoritması için kullanılacak görevlerin belirlenmesi gerekir.

- i. Aşağıda gösterildiği gibi sağ tarafında duvar varken robot düz bir şekilde ilerler. Bunu yaparken ne sağdaki duvara çok yaklaşmalıdır ne de ondan fazlaca uzaklaşmalıdır.



Resim 137. Robotun Düz İlerlemesi

- ii. Aşağıdaki şekilde görüldüğü üzere hem karşısında hem de sağında duvar varsa robot sola dönmelidir.

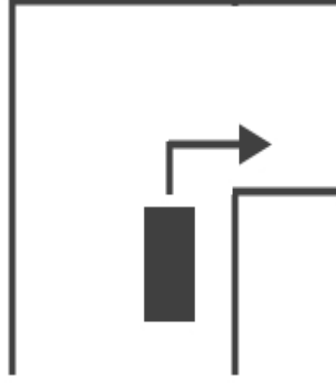


Resim 138. Robotun Sola Dönmesi

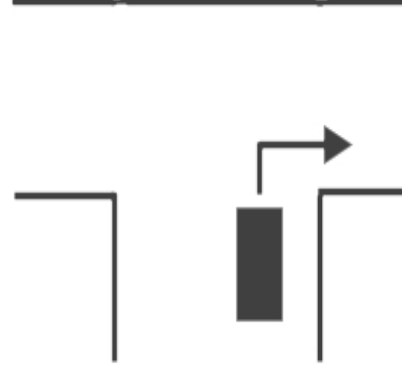
- iii. Duvarın sağ tarafında boşluk olduğu durumlarda robot sağa dönmelidir. Aşağıdaki resimlerde bu durumlar görülebilir.



Resim 139. Sağa Dönme Durum 1



Resim 140. Sağa Dönme Durum 2



Resim 141. Sağa Dönme Durum 3

1.4. Robot Gereksinimleri

Bu etkinlikte robotun ileriye ve sağ tarafına bakacak şekilde iki tane mesafe sensörü olması gerekir. İleriye bakan sensör karşı duvarı algılamak için kullanılır. Sağa bakan sensör ise robotun sağ duvarı takip etmesi ve sağ duvarda bir açıklık olup olmadığını belirlemesi için kullanılır.



Resim 142. Robot Tasarımı

1.5. Alt Problemler

Sağ duvar algoritmasını kullanarak labirentten çıkma probleminin alt problemleri belirlenmelidir. Alt problemler aşağıdaki şekildedir:

- i. Robot sağ duvara ne çok yakın ne de çok uzak olacak şekilde ileriye doğru gitmelidir (duvar takip),
- ii. Karşıda bir duvar varsa ve sağ duvarda açıklık yoksa robot sola dönmelidir (sola dönme),
- iii. Sağ duvarda bir açıklık varsa robot sağa dönmelidir (sağa dönme),
- iv. i, ii ve iii numaralı adımlardaki işlemler sürekli tekrarlanmalıdır (tekrarlama).

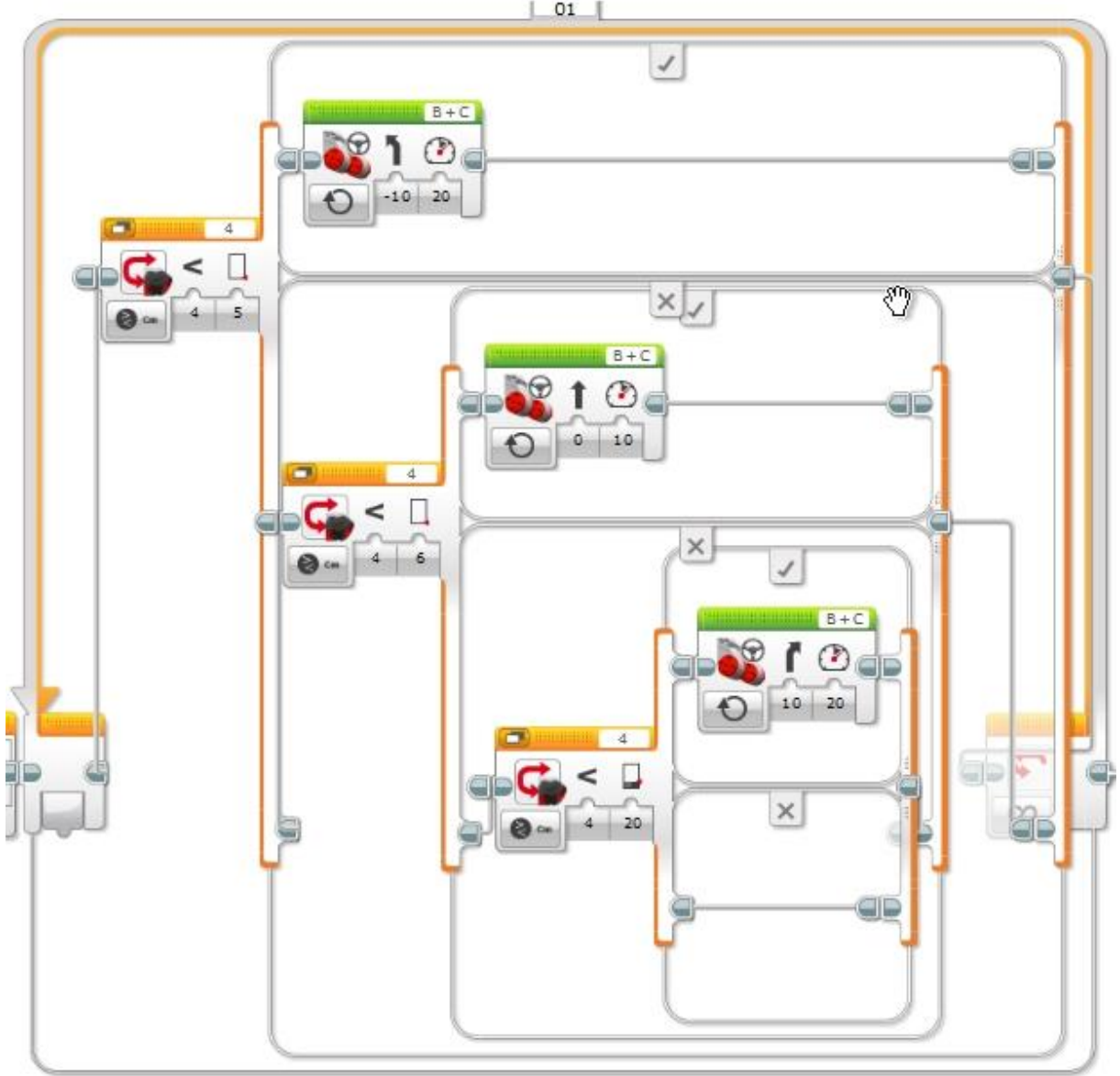
2. UYGULA

2.1. Duvar Takip

Bu görevde robotun sağındaki duvarı yaklaşık olarak 6 cm mesafede takip etmesi gerekir. 6 cm'lik mesafenin sürekli sabit kalması zorunlu değildir. Bu bazen 9, bazen 3 olabilir. Yani belirli bir aralıkta olmak kaydıyla robot sağdaki duvarı takip etmelidir. Bu görev için 4 numaralı porta takılmış olan (sağa bakan) mesafe sensörü kullanılır. Duvar takip algoritması şu şekilde özetlenebilir:

- i. Robot 5 cm'den daha fazla duvara yaklaştıysa sola doğru 20 hızında 10 derece hareket ettirilir,
- ii. Robotun duvara olan mesafesi 5 ve 6 cm arasındaysa robot 10 hızında ileri doğru gider,
- iii. Robotun duvara olan mesafesi 6 cm'den fazla 20 cm'den az ise robot 20 hızında sağa doğru 10 derece hareket ettirilir.

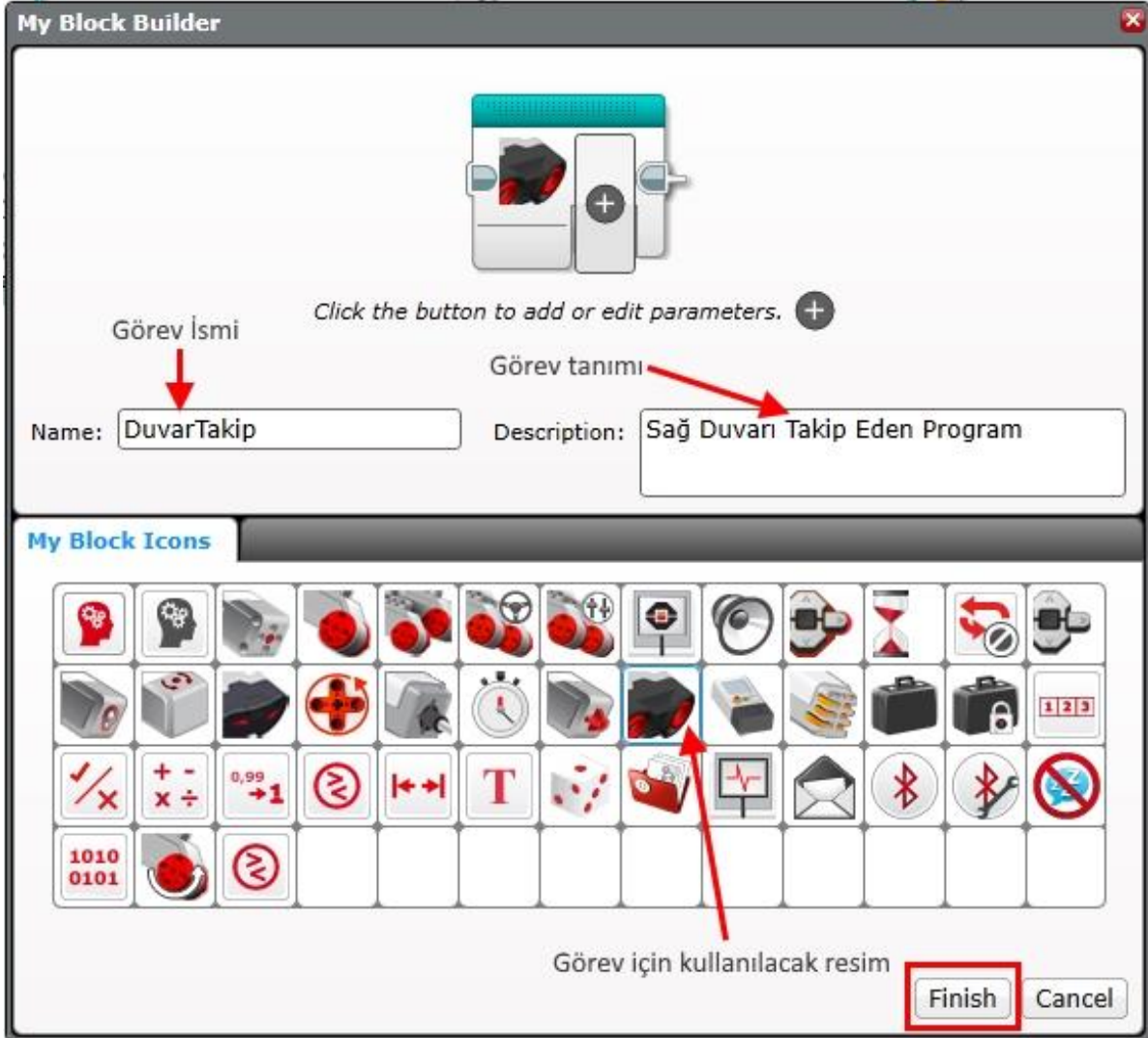
Görevin kodu aşağıdaki gibidir. Bu kod ile robotun sağda bulunan duvarı takip etmesi sağlanır. Robot sağ duvara yakınsa ($\text{mesafe} < 5 \text{ cm}$) sola doğru yönlendirilir. Sağ duvar ile arasındaki mesafe yeterli ise ($5 \text{ cm} < \text{mesafe} < 6 \text{ cm}$) düz gitmesi sağlanır. Eğer sağ duvardan uzaklaşmışsa robot sağa doğru yönlendirilir ($6 \text{ cm} < \text{mesafe} < 20 \text{ cm}$). Burada kod içinde kullanılan 20 cm sınırının açıklanması yerinde olacaktır. Duvarda bir açıklığın bulunduğunu belirlemek için eşik değeri 20 cm olarak belirlenmiştir. 20 cm'ye kadar olan açıklıklarda duvarda bir boşluk olmadığı kabul edilir. Aşağıda bulunan kod herhangi bir duvar kullanılarak çalıştırılır ve öğrencilerle birlikte sonuçlar gözlemlenir.



Resim 143. Duvar Takip Programı

Bu kod bir blok hâline getirilmelidir. Döngüdeki kodların tamamı seçilir. Bu aşamada öğrenciler döngünün neden bloğa dâhil olmadığını merak edebilir. Döngünün içerisine daha sonra sola ve sağa dönme komutları da konulacağı için döngünün bloğun içerisine dâhil edilmediği öğrencilere açıklanır.

Araçlar (Tools) menüsünden My Block Builder alt menüsü seçilir. Açılan pencereden görevin ismi ve tanımı (yazılması zorunlu değildir) yazılır. Ardından görev bloğunun üzerinde görünecek olan görev resmi seçilir ve Bitir (Finish) butonuna tıklanarak blok oluşturulur. Oluşturulan bloğa istenildiğinde “Bloklarım (My Blocks)” sekmesinden ulaşılabilir.



Resim 144. Duvar Takip Bloğu

2.2. Sola Dönme

Bu görevde robot (sağ duvarda bir açıklık yokken) karşısında bir duvar gördüğünde sola döner. Bunun için 3 numaralı porta bağlanan (ileriye bakan) mesafe sensörü kullanılır. Bu görev şu şekilde özetlenir:

Karşısında bir engel algıladığında:

- i. Robot durur,
- ii. Robot (dönüşte duvara çarpmamak için) bir miktar geri gelir,
- iii. Robot sola döner.

Bu görevin kodu aşağıdaki gibidir. Bu kod çalıştırılarak öğrencilerle birlikte test edilir.

Dikkat: Bu görev, döngüde duvar takip programının sağına konulacaktır. Bu nedenle bir döngü içinde gösterilmemiştir. Kod döngüye yazılarak test edilmelidir.



Resim 145. Sola Dönme Programı

Göreve SolaDon ismi verilir ve Araçlar (Tools) menüsünden My Block Builder kullanılarak blok hâline getirilir.

2.3. Sağa Dönme

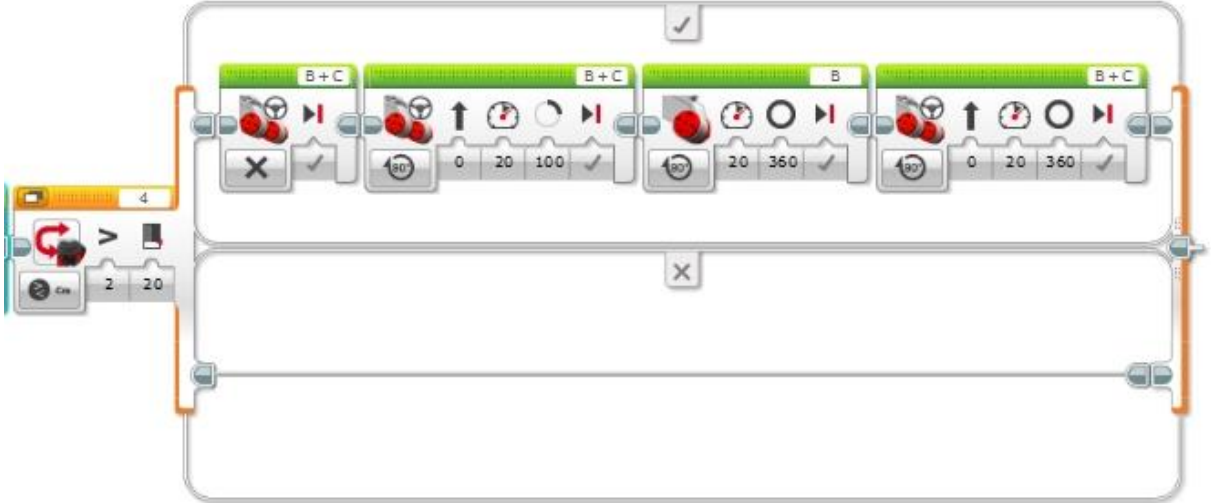
Bu görevde robot sağ duvarda bir boşluk algıladığında sağa döner. Bunun için sağa bakan ve 4 numaralı porta bağlı olan mesafe sensörü kullanılır. Bu görev şu şekilde özetlenir:

Sağ tarafta bir boşluk algıladığında:

- (i) Robot durur,
- (ii) Robot (dönerken duvara çarpmamak için) bir miktar ilerler,
- (iii) Robot sağa döner,
- (iv) Robot (yeni koridora girmesini sağlamak için) bir miktar ilerler.

Bu görevin kodu aşağıdaki gibidir. Rehber öğretmen bu kodu çalıştırarak öğrencilerle birlikte test eder.

Dikkat: Bu görev ana döngüye yerleştirilir. Bu nedenle bir döngü içinde gösterilmemiştir. Kod bir döngüye yazılarak test edilmelidir.

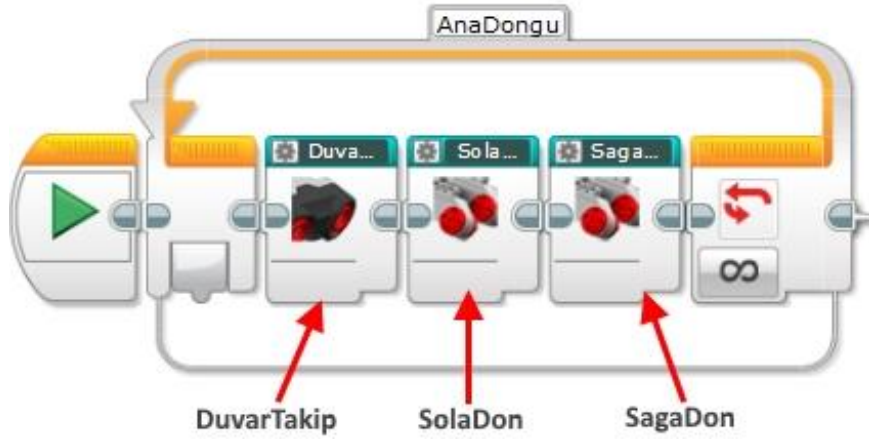


Resim 146. Sağa Dönme Programı

Göreve SagaDon ismi verilir ve Araçlar (Tools) menüsünden My Block Builder kullanılarak blok hâline getirilir.

2.4. Görevlerin Birleştirilmesi

Problem üç alt probleme bölünerek "duvarı takip et", "sola dön" ve "sağa dön" olmak üzere üç farklı görev tanımlanmış; bu görevlerin kodu yazılarak blok hâline getirilmiştir. Oluşturulan alt çözümler, yani bloklar birleştirilince ana problem de çözülmüş olacaktır. Bu aşamada söz konusu problem için çözüm oldukça basittir ve kodu aşağıdaki şekildedir.



Resim 147. Blokların Birleştirilmesi

Dikkat

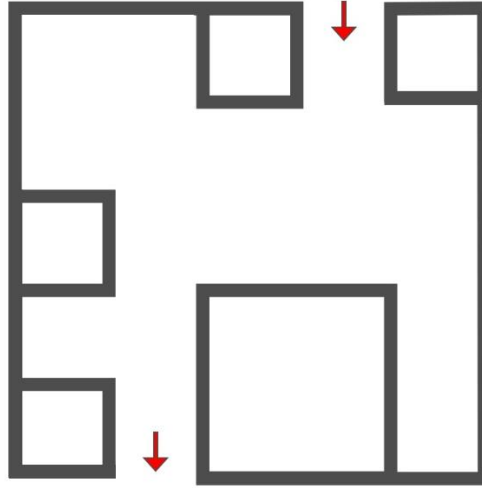
Yukarıda hazırlanan program her ortamda çalışmayabilir. Bunun nedeni, kullanılan algorithmada hata bulunması değildir. **Algoritma doğrudur ve daha önce denenmiştir.** Programın doğru çalışması için yazılan kodda bulunan

parametreler sınıf ortamında bulunan koşullara göre uyarlanmalıdır. Rehber öğretmen kodların kendi durumlarında çalışması için kod içinde bulunan parametreleri düzenlemelidir.

3. TASARLA VE ÜRET

3.1. Labirent Yarışması

Derste kullanılacak labirent aşağıdadır. **Labirentin koridorlarının genişliği 30 cm'dir.** Rehber öğretmen labirenti oluşturulurken merkezlerde bulunan 20 cm x 30 cm boyutundaki mukavvalar ve rehber öğretmen tarafından üç boyutlu yazıcıda bastırılacak birleştirme aparatları kullanılacaktır. Birleştirme aparatlarının çizimi haftanın belgeleri içinde bulunmaktadır. Birleştirme aparatları kullanılarak mukavvalar birleştirilir ve labirent oluşturulur. **Rehber öğretmen dersten önce gerekli hazırlıkları yapmalıdır.**



Resim 148. Labirent

Yarışmanın kuralları:

- Yarışma için her öğrenci grubuna 45 dakikalık süre ve labirenti iki kere kullanma hakkı verilir.
- Öğrencilerin yaptıkları iki denemeden az süreli olanı grubun puanı olarak değerlendirilir.
- Öğrenciler ilk kullanımın ardından kodlarında veya tasarımlarında değişiklikler yapabilirler. Fakat labirentin ikinci defa kullanımı son haklarıdır ve bundan sonra değişiklik yapamazlar.
- En düşük puana sahip olan grup, yani labirenti en kısa sürede çözen grup yarışmayı kazanır.
- Robot labirenti çözerken duvara çarpmamalıdır. Robotu duvara çarpan grup labirent kullanma hakkını kaybeder. Eğer o grup ikinci labirent kullanma hakkında da robotu duvara çarptırırsa yarışmadan elenir.

- Gruplar denemeler yapmak için öğretmenin sağladığı labirent dışında sınıfta bulunan nesnelere kullanabilir. Örneğin sınıf duvarını kullanarak duvar takip kodunu test edebilirler.

3.2. Tasarla

Fiziksel olarak robotu oluşturma ve programlama adımlarını yazma sürecinden önce tanımlama ve fikir üretme sürecinin gerçekleştirilmesi önemlidir.

- **Tanımlama Süreci:** Öğrencilerin robotun ve programın neler gerektirdiğini belirlemesi/ortaya koyması gerekir. Öğrencilerin robot ve program için ayrı ayrı işlemleri maddeler hâlinde yazması beklenir.
- **Fikir Üretme Süreci:** Bu aşamada, öğrencilerin tanımlama sürecinde belirlenen işlemlerin nasıl yapılabileceği ile ilgili fikir yürütmesi gerekmektedir. Öğrencilerin robotun oluşturulması ve program kodlarının yazılması sürecinde çözüm için sundukları fikirleri maddeler hâlinde belirtmeleri önemlidir.

3.3. Üret

Öğrencilerin önce, tanımladıkları ve hakkında fikir ürettikleri robotu oluşturmaları gerekir. Robotun istenen şekilde ve sorunsuz hareket edip etmediği test edilmeli, sorunlar varsa gözden geçirilip düzeltilmelidir. Daha sonra program tamamlanmalı ve kontrol edilip eksiklikleri giderilmelidir.

Dikkat: Örnek çözüme rehber öğretmenlere verilen dosyalardan ulaşılabilir. Verilen çözüm, problemi çözmek için tek yol değildir. Öğrenciler birçok şekilde ilerleyerek kendi kodlarını geliştirebilir.

3.4. Labirent Çözümünün İyileştirilmesi

Yazılan kodların iyi çalışıp çalışmadığını öğrenciler ve rehber öğretmen gözlemlemelidir. Robot bazen labirentten çıksa da belirli durumlarda labirentten çıkamamaktadır ya da bazı görevleri yerine getirirken bazılarını yerine getirememektedir. Bunun nedeni alt problemler için geliştirilen çözümlerin tam olarak yeterli olmamasıdır. Bu çözümlerden bazıları iyileştirilebilir. Öğrenciler iki şekilde iyileştirme yapabilirler:

- (i) Yazılan kodlardan bazıları yeterli hassasiyete sahip değildir. Bu kodlar geliştirilebilir.
- (ii) Robot fiziksel olarak fazla yer kaplamaktadır. Robot başka bir tasarımla küçültülebilir.

İpuçları (Bunları öğrencilere başta söylemek uygun değildir, öncelikle kendilerinin keşfetmesine veya sorarak öğrenmelerine izin/ mkân verilmelidir):

- **İpucu 1:** Robot duvara çok yaklaştığında ve duvardan çok uzaklaştığında ani tepki veremediği için rotasından fazlaca sapar.
- **İpucu 2:** Robot ile duvar arasındaki mesafe 3 cm'nin altına düştüğünde mesafe sensöründen gelen değer birden 255'e çıkar. Bu durum, mesafe sensörünün

yakın değerler için yeterince iyi olmadığını gösterir. Fakat bu durum dikkate alınarak yazılan kod iyileştirilebilir.

- **İpucu 3:** Robot tasarımında akıllı tuğla yatay durumda kullanılır. Akıllı tuğla dikey hâle getirilerek robotun daha az yer kaplaması sağlanabilir.

3.5. Labirent Görevinin Zorlaştırılması

Önceki görevleri bitiren öğrenciler labirentin koridor genişliğini 20 cm yaparak robotun 20 cm'lik labirentten çıkması üzerinde çalışabilir.

4. DEĞERLENDİR

Bu aşamada hedef, öğrencilerin öğrenme sürecinde yaşadıkları ve öğrendikleri üzerine düşüncelerini sağlamaktır. Bu sayede öğrenciler; problem çözme yetenekleri, dersin konusu ve kendileri ile ilgili gözlemler yaparak öğrendikleri yeni konuları ve kendilerini değerlendirmekle beraber sonraki çalışmalarını planlamak için de fırsat elde edeceklerdir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- En çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?) Öğrenciler aşağıdaki problemleri bildirmezlerse rehber öğretmen bu problemleri nasıl çözdüklerini sorabilir.
 - Duvar takip etme görevinde zorluklar yaşadınız mı?
 - Sola ve sağa dönme görevinde zorluklar yaşadınız mı?
 - Birleştirme görevinde zorluklar yaşadınız mı?
- Problemleri çözerken ne gibi sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla anlaşmazlığa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan ne/neler öğrendiniz?
- Bu hafta öğrendiğiniz konuları günlük hayatta nerelerde uygulayabilirsiniz?

Değerlendirme, öğrencileri sıkmadan, her soru için verilen cevaplar tatmin edici bir düzeye ulaşınca kadar devam ettirilir.

Proje Hazırlıyorum

Bu haftada, proje çalışmalarına devam edilmesi önerilmektedir. Projenin “fikir üretme” süreci için planlamalar yapılmalıdır. Detaylar için EK'ler incelenebilir.

12. Hafta: Bluetooth

Ön Bilgi:

- Öğrenciler, robot kavramını temel düzeyde bilir.
- Öğrenciler, robot setiyle farklı robot tasarımları yapmıştır.
- Öğrenciler, robot setini programlamak için EV3 yazılımını kullanmıştır.
- Öğrenciler, robotun hareket etmesi için gerekli programlama adımlarını oluşturmuştur.
- Öğrenciler, sensörlerin kullanıldığı programlar yapmıştır.
- Öğrenciler, kablo (wire) ile bloklar arası veri iletimini kullanmıştır.
- Öğrenciler, EV3 yazılımında veriler üzerinde matematik ve mantık işlemlerini gerçekleştirmiştir.

Haftanın Kazanımları:

- Öğrenciler EV3 tuğlanın *bluetooth* özelliğini kullanır.
- Öğrenciler EV3 tuğlalar arasında mesaj gönderip alabilir.

Haftanın Amacı:

Bu haftanın amacı, öğrencilerin EV3 akıllı tuğlada bulunan *bluetooth* özelliğini kullanarak iki akıllı tuğla arasında bağlantı gerçekleştirmelerini sağlamaktır. Ayrıca bağlı olan iki EV3 akıllı tuğla arasında veri alışverişini gerçekleştirerek bir robotun diğer robot ile yönlendirilmesini sağlamaktır.

Kullanılacak Malzemeler:

Robot seti, bilgisayar.

Haftanın İşlenişi:

Gözle: İki farklı akıllı tuğla arasında *bluetooth* bağlantısını yapma ve tuğlaları eşleştirme, *bluetooth* bağlantısı kullanılarak tuğlalar arasında mesajlaşmayı sağlama ve bir robotun okuduğu sensör verilerini diğer robota aktarma.

Uygula: Bluetooth Bağlantısı (Bluetooth Connection) ve Mesajlaşma (Messaging) bloklarını uygulama.

Tasarla: Birbirine *bluetooth* bağlantısı ile bağlı iki robottan birinin yaptığı hareketlerin diğer robot tarafında taklit edilmesi için gerekli programı tanımlama ve planlama.

Üret: İstenen taklitçi robot programı üretme.

Değerlendir: Haftanın içeriği ile ilgili yansıtma etkinliği.

1. GÖZLE VE UYGULA

1.1. Gözle: Bluetooth

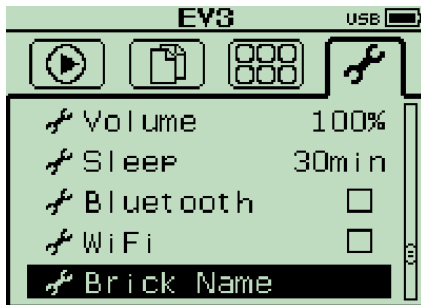
Öğrencilere *bluetooth* bağlantısı hakkında bilgilerinin olup olmadığı sorulur. Öğrenciler konuyla ilgili dinlendikten sonra, öğrencilere *bluetooth* bağlantısı kullanarak robotlarının birbiriyle haberleşmelerini sağlayacakları belirtilir. Yapılacak etkinliklerde iki adet robot gerektiği için ikişerli gruplar (4 öğrenci) hâlinde çalışacakları ifade edilir.

Rehber öğretmen, *bluetooth* teknolojisinin verilerin kısa mesafede kablosuz olarak aktarılmasını sağlayan bir teknoloji olduğunu belirtir. Açık alanda 10 metreye kadar çekim alanına sahip olduğunu ifade eder. 10 metre, veri aktarımı ve haberleşme için oldukça kısa bir mesafe olsa da günümüzde *bluetooth* teknolojisinin kullanıldığı birçok alan bulunduğunu ifade eder ve öğrencilerden *bluetooth*un kullanım alanlarına örnekler vermelerini ister. Aşağıda birkaç örnek verilmiştir.

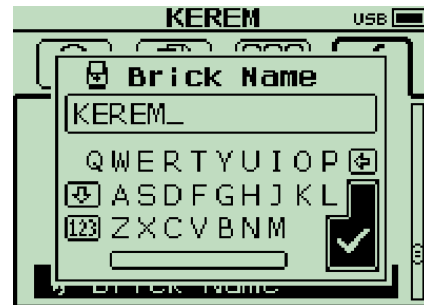
- Bluetooth kulaklık
- Bluetooth fare ve klavye
- Akıllı kol saatleri
- Bluetooth hoparlör
- Bluetooth otomobil sistemleri

Bluetooth ile cihazlar arasında haberleşmenin sağlanabilmesi için, öncelikle iki cihazın birbiriyle eşleştirilmesi gerekir. Eşleştirme öncesinde her iki cihazın *bluetooth* özellikleri aktif hâle getirilir. İki cihazın birbiriyle eşleştirilmesi esnasında girilen parola ile bağlantının güvenliği sağlanır.

Etkinliklerde robotların birbiriyle eşleştirilmesi esnasında herhangi bir sorun yaşanmaması için öncelikle tüm gruplardan robotlarına isim vermeleri istenir. Öğrenciler robotlarının isimlerini akıllı tuğla üzerindeki düğmeleri kullanarak ayarlar (settings) menüsündeki Tuğla İsmi (Brick Name) alt menüsünden değiştirebilirler. Ekranda açılan klavye üzerinde akıllı tuğla düğmeleriyle istenen karakterin üzerine gelinerek tuğlanın ismi değiştirilir.

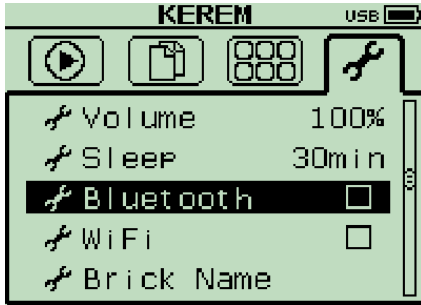


Resim 149. İsim Değiştirme Adım 1

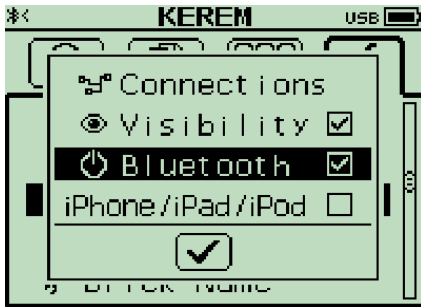


Resim 150. İsim Değiştirme Adım 2

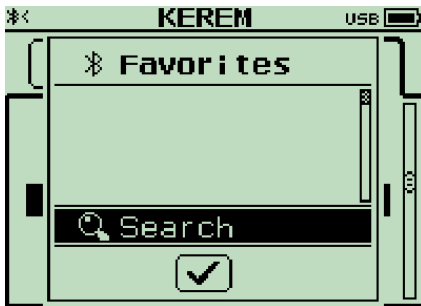
Robotların birbiriyle iletişimi geçebilmesi için *bluetooth* bağlantılarının aktif hâle getirilmesi ve birbiriyle eşleştirilmesi gerekmektedir. Robotları eşleştirmek için aşağıdaki adımlar takip edilir.



Ayarlar menüsünden Bluetooth seçeneğini seçilir.



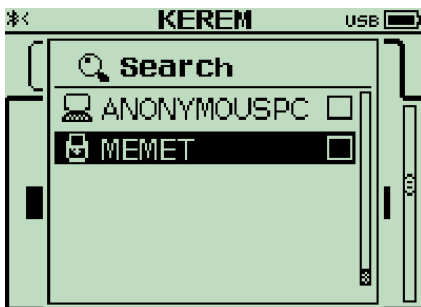
Açılan menüde Bluetooth seçeneği aktif hâle getirilir. Görülebilirlik (Visibility) seçeneği de aktif hâle getirilir. Sonra Bağlantılar (Connections) seçeneğine gidilir.



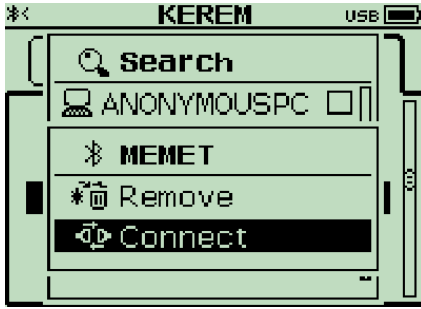
Ara (Search) seçilerek yakında bulunan diğer *bluetooth* cihazların listelenmesi beklenir.



Arama süreci birkaç dakika sürebilir.



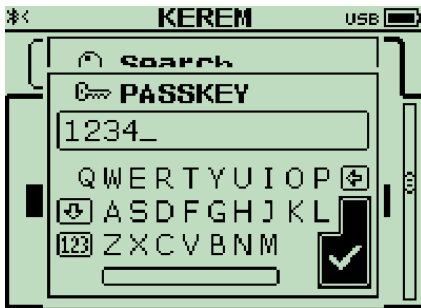
Bu süreçte robotun çevresinde bulunan cihazların isimleri sıralanır. Robotun eşleştirilmek istendiği cihaz, ismi üzerine gelinerek seçilir.



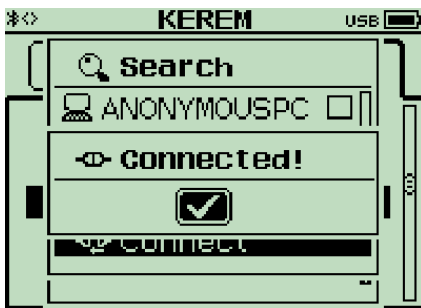
Seçilen cihaza bağlanmak için Bağlan (Connect) seçeneği seçilir.



Bağlanma isteği doğrulanır.



Akıllı tuğla üzerindeki düğmeler kullanılarak bir şifre belirlenir (1234 şifresi kullanılabilir) ve Giriş (Enter) tuşu işaretlenir. Eşleştirilmek istenen robotun ekranında da şifre onayı belirecektir, aynı şifre yeniden yazılarak Giriş (Enter) tuşu işaretlenir.



Ekranda Bağlandı (Connected!) yazısı ve bağlantı sesi duyulduğunda bağlantı işlemi tamamlanmış demektir.

Not

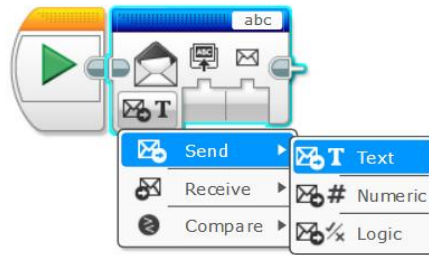
Aynı ortamda çok sayıda *bluetooth* cihazının bulunması, bağlanma süresini artırmakta ve bağlantı sorunlarına sebep olabilmektedir. Grupların, robotlarını ayrı bir ortamda eşleştirmeleri tavsiye edilir.

1.2. Gözle: Bluetooth Bağlantısı ve Mesajlaşma Blokları



Resim 151. Bluetooth Connection/Bağlantı Bloğu

Bluetooth Bağlantısı (Bluetooth Connection) bloğu EV3 yazılımında mavi İleri Düzey (Advanced) sekmesinde yer alır. Bluetooth Connection bloğunun dört modu bulunur. Bu blok ile akıllı tuğlanın *bluetoothu* açılabilir (On) veya açık olan *bluetooth* kapatılabilir (Off). Robot “Başlatma (Initiate)” moduyla belirtilen başka bir robot ile (bloğun sağ alt köşesinden, eşleştirilmek istenen robotun ismi yazılarak) eşleştirilebilir. “Temizle (Clear)” modu ise eşleştirilen robotla kurulan *bluetooth* bağlantısının sonlandırılması için kullanılır.



Resim 152. Bluetooth Messaging/Mesajlaşma Bloğu Seçenekleri

Öncesinde eşleştirilmiş iki akıllı tuğlanın birbiriyle haberleşmesini sağlamak üzere mavi sekmede yer alan Mesajlaşma (Messaging) bloğu kullanılır. Mesajlaşma bloğu kullanılarak mesaj gönderilebilir (Send), alınabilir (Receive) ve alınan mesaj var olan bir değer ile karşılaştırılabilir (Compare). Ayrıca gönderilen veya alınan mesaj Metin (Text), Sayısal (Numeric) ve Mantık (Logic) türlerinde olabilir.

Bir akıllı tuğladan diğerine mesaj gönderebilmek için üç parametrenin belirlenmiş olması gerekir. Bunlar:

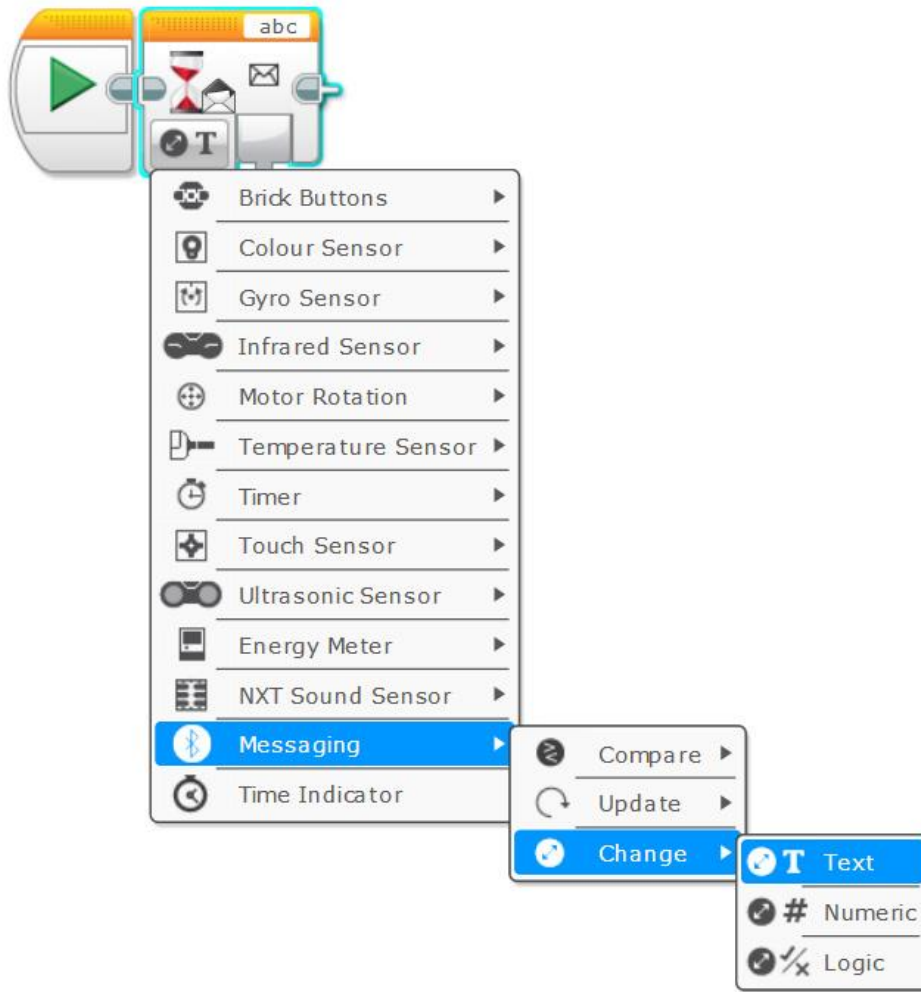
- **Mesaj başlığı:** Mesajlaşma bloğunun sağ üst köşesinde yer alır.
- **Mesajın iletileceği akıllı tuğlanın ismi:** Bluetooth bloğu kullanılarak eşleştirilen diğer akıllı tuğlanın ismi.
- **İletilecek mesajın içeriği:** Aktarılacak mesajın türüne bağlı olarak metin, sayı veya mantık (Doğru/Yanlış) değeri olabilir. Elle değer girilebileceği gibi kablo ile başka bir bloktan da aktarılmış olabilir.

Benzer şekilde, eşleştirilen tuğlanın gönderilen mesajı alması için yine Mesajlaşma (Messaging) bloğu kullanılır fakat Alma (Receive) modunun seçilmiş olması

gerekmektedir. Mesajın alınabilmesi için Gönder (Send) bloğunda belirtilen mesaj başlığı ve mesaj veri türünün (Text, Numeric, Logic) iki robotta da aynı olarak belirlenmesi gerekir.

Ayrıca, Mesajlaşma bloğu ile alınan mesajın içeriğinin var olan bir değer ile kıyaslanması da Karşılaştır (Compare) modu kullanılarak yapılabilir. Karşılaştırma işleminin sonucunda doğru veya yanlış olmak üzere Mantık (Logic) veri türünde bir sonuç elde edilir. Bu sonuç kablo ile bir diğer bloğa aktarılabilir. Özellikle Anahtar (Switch) bloğuna aktarılacak ise bu aşamada karşılaştır (compare) modunu kullanmak anlamlı olabilir.

Turuncu sekmede yer alan “bekle (wait)” bloğunun herhangi bir *bluetooth* mesaj gelinceye kadar programı bekletmek amacıyla kullanılabilen de unutulmamalıdır.



Resim 153. Wait Bloğu bluetooth İşlemleri

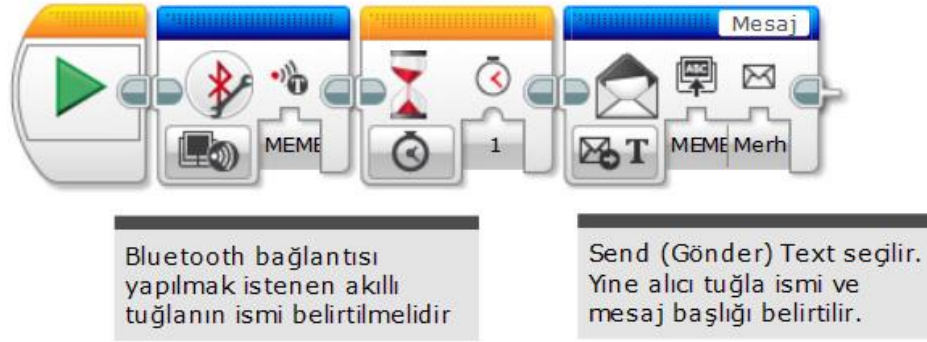
1.3. Uygula: Merhaba Mesajı

Öğrencilere iki grubun birleşerek birlikte çalışacakları belirtilir, böylece her yeni grupta iki adet akıllı tuğla olacaktır. Öğrencilerden her bir gruba ait iki robotu *bluetooth* ile birbirine bağlamaları istenir. Sonrasında bir robottan gönderilecek mesajın diğer robotun ekranında yazdırılmasını sağlayacak programları oluşturmaları istenir.

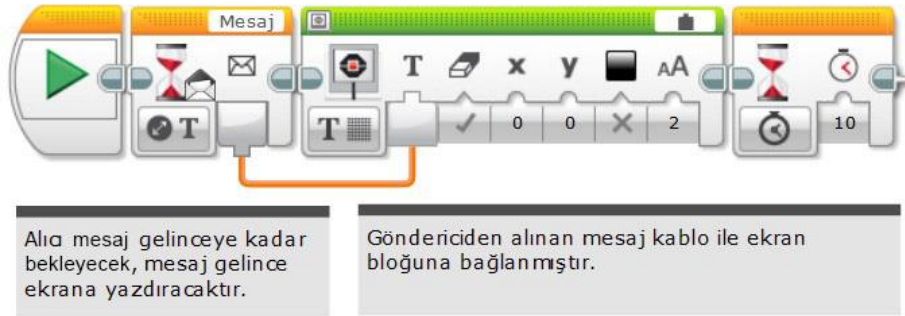
Öğrencilerin deneme yanılma yöntemini kullanmalarına imkân tanınır. Programlarında oluşacak sorunları tartışmaları ve çözüm önerileri üretmeleri desteklenir. Özellikle robotların eş zamanlı haberleşmelerini nasıl sağlayacakları konusunda çözümler üretmeleri istenir.

Bir robot mesajı göndereceği, diğer robot da alacağı için iki robotta farklı program çalıştırılır. Gün içinde öğrencilerin, görevleri dönüşümlü olarak gerçekleştirmeleri istenir.

Örnek gönderici ve alıcı programları aşağıdadır.



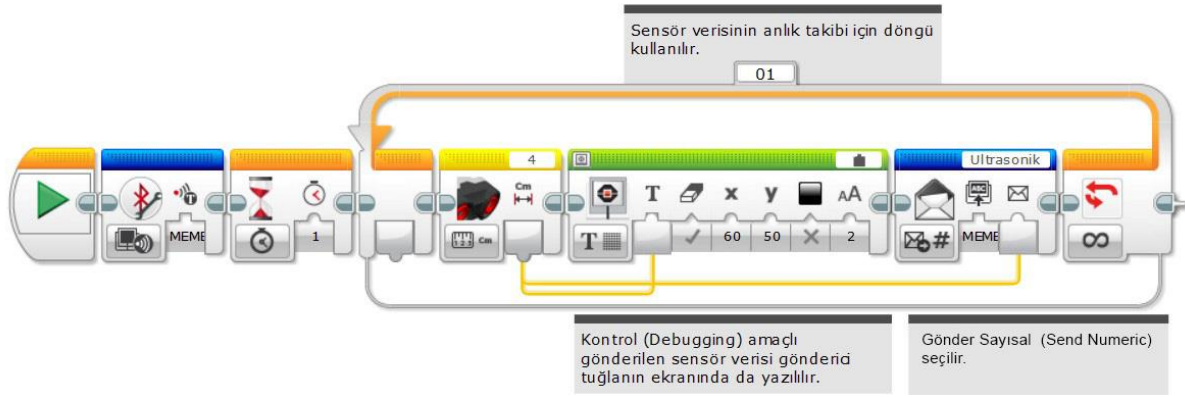
Resim 154. Örnek Gönderici Program



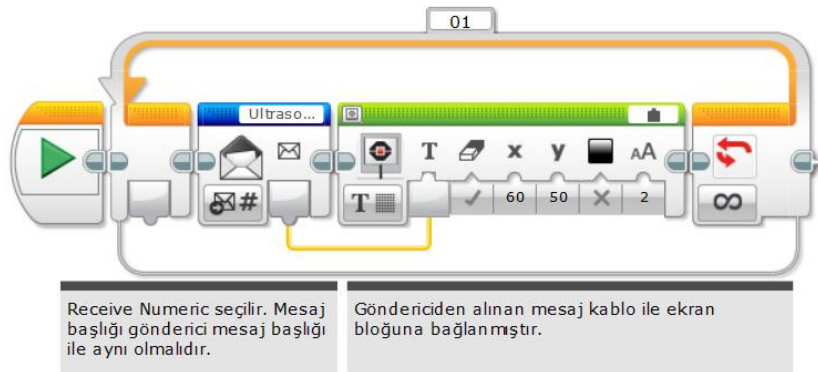
Resim 155. Örnek Alıcı Program

1.4. Uygula: Sensör Verilerinin Diğer Robota Aktarılması

Öğrencilerden, bir robotun algıladığı sensör verilerinin diğer robotun ekranında görünmesini sağlayan programı yapmaları istenir. Bu amaçla ultrasonik sensörün ölçtüğü mesafenin diğer robotun ekranında görünmesi istenir. Robot bir engele yaklaştıkça diğer robotun ekranındaki değerlerin de değişmesi gerekir. Yani ölçüm sonucu ekrana yalnızca bir defa yazdırılmamalı, sürekli olarak güncellenmelidir. Örnek program aşağıda verilmiştir.



Resim 156. Örnek Gönderici Program



Resim 157. Örnek Alıcı Program

Öğrencilerden iki farklı sensöre ait verilerin, eşleştirilen robota nasıl aktarılabilceğini tartışmaları istenir. Yeterli vaktin olması durumunda hem mesafe sensörünün ölçtüğü uzaklık verilerin hem de renk sensörünün ölçtüğü yansıyan ışık verilerinin diğer robotun ekranına yazdırılması istenebilir (**İpucu:** Her bir sensör için farklı mesaj başlığı kullanılır).

2. TASARLA

2.1. Tasarla: Taklitçi Robot

Bu etkinlikte öğrencilerden bir robot bağımsız olarak hareket ederken diğer robotun aynı hareketi gerçekleştirmesini sağlayacak programı hazırlamaları istenir (örneğin, önceki haftalarda gerçekleştirilen, öndeki aracı takip eden veya çizgiyi takip eden robot etkinlikleri).

Tasarla: Öğrencilerden gün içinde öğrendikleri "bluetooth bağlantısı ile robotların haberleşmeleri" yöntemini kullanarak, bir robotun yaptığı hareketin aynısını yapan taklitçi bir robot tasarlama ve programlamaları istenir. Birinci robotun tamamen bağımsız olduğu, taklitçi robotun ise birinci robota bağlı olduğu belirtilir. Birinci robotun hareketi önceden belirlenmeyecektir, robot ortama uygun herhangi bir hareketi yapacaktır (birinci robot elle de hareket ettirilebilir). Sadece ileri geri değil, sağ sola dönme hareketleri de taklit edilmelidir (Örneğin, çizgi izleyen robot etkinliğindeki robotun hareketi).

Tanımlama: Öğrenciler öncelikli olarak problemi tanımlamalıdır. Problemi çözmelerine yardımcı olacak şu soruların cevaplarını kendi aralarında tartışsınlar:

- Taklitçi robotun taklit etmesi istenen nedir?
- Taklitçi robot, birinci robottan hangi verileri almalıdır?
- *Bluetooth* üzerinden kaç farklı veri türü paylaşılacaktır?
- Sağa veya sola dönme hareketleri nasıl taklit edilebilir?

Fikir Üretme: Bu aşamada, öğrencilerin yukarıda belirlenen işlemleri robotun nasıl gerçekleştirebileceği ile ilgili fikir yürütmesi gerekir. Öğrenciler aşağıda belirtilen çözüm önerilerine benzer fikirler üretebilirler.

- *Bluetooth* üzerinden hareketi sağlayan B ve C motorlarının verileri paylaşılmalıdır.
- Sağa veya sola dönüşler için B ve C motorlarının verilerinin bağımsız olarak paylaşılması gerekir.
- Hareketin eş zamanlı olması için döngü kullanılması gerekir.
- Taklitçi robotun B ve C motor açıları sürekli olarak, birinci robotun B ve C motorunu açısına eşitlenmelidir.

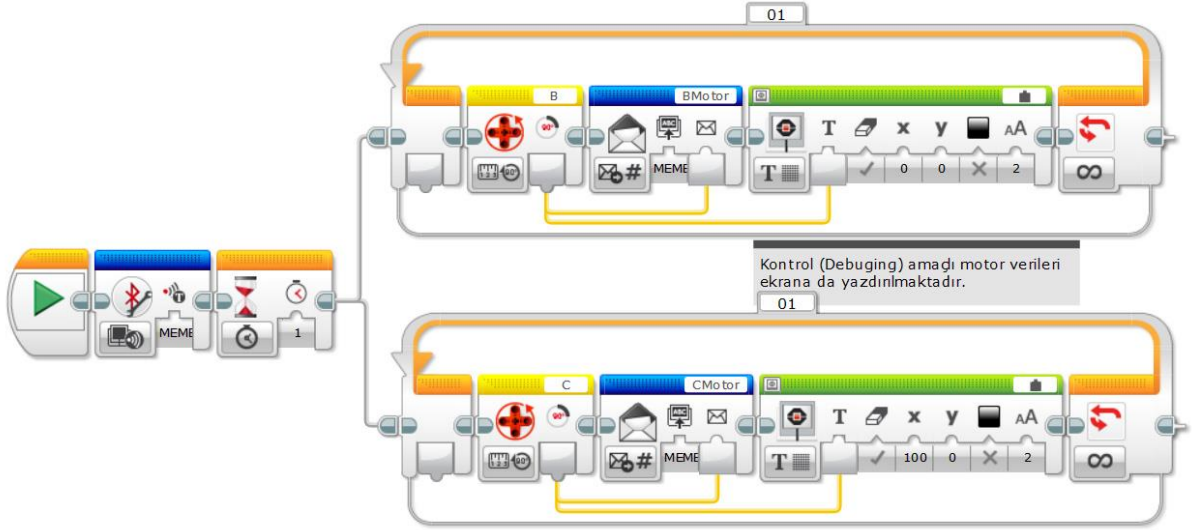
B ve C motorlarının başka hangi verileri paylaşılabilir?

Not

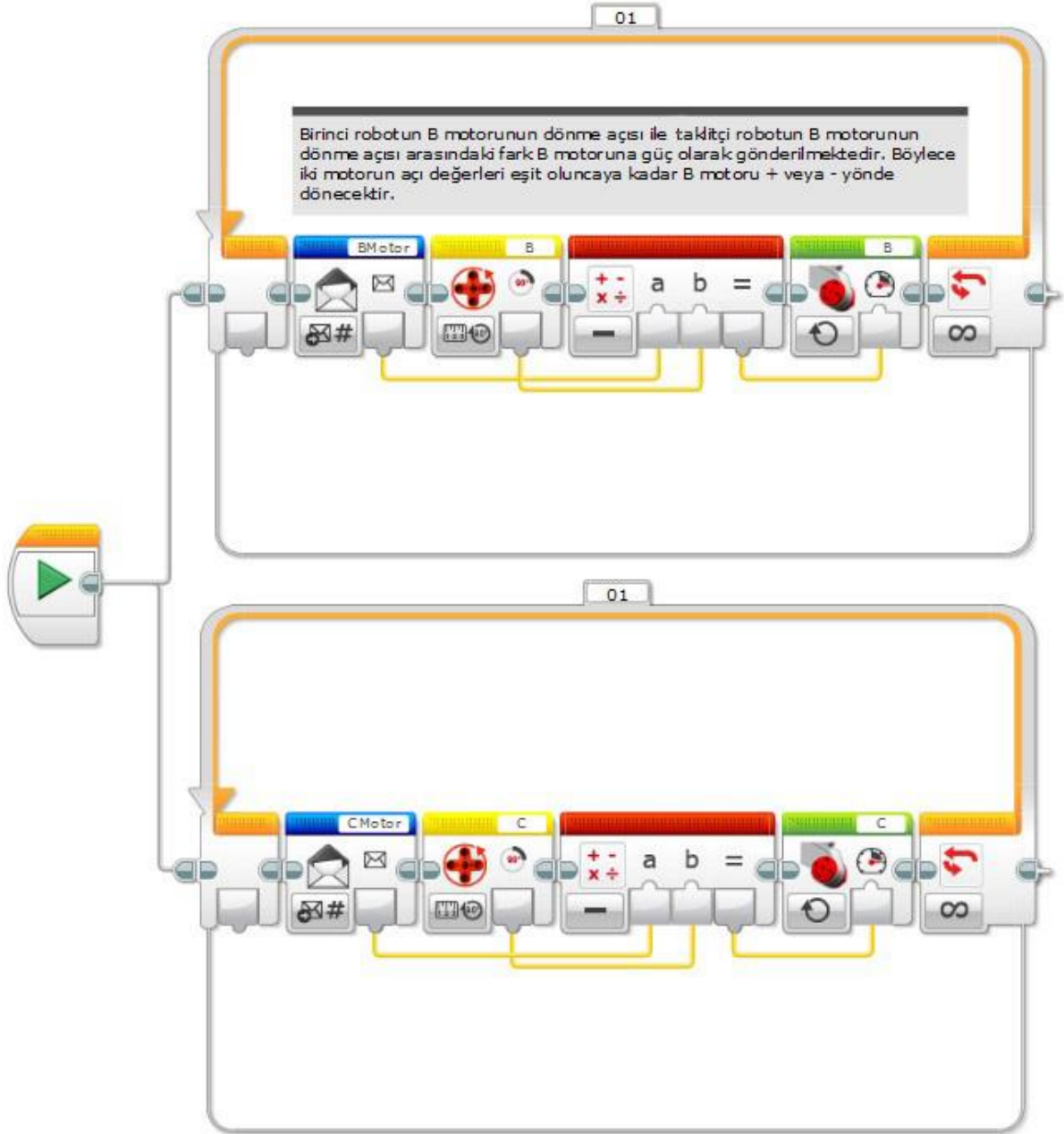
Öğrenciler bu süreçte, sadece B ve C motor verilerinin iletilmesinin haricinde, mesafe, renk veya dönme sensörü verilerinin iletilmesi ile taklitçi robotun hareketini belirleme yolunu seçebilirler. Problemin çözümünde tek yöntemin olmadığı unutulmamalıdır. Mantıklı çözüm önerileri desteklenmelidir.

3. ÜRET

Öğrenciler çözüm tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenen görevi yerine getirirler. Birinci ve taklitçi robot için örnek programlar aşağıda sunulmuştur.



Resim 158. Birinci Robot Örnek Program



Resim 159. Taklitçi Robot Örnek Program

Not

Benzer program, birinci robotun B ve C motorlarından ölçülen Mevcut Güç (Current Power) değerinin, taklitçi robotun ilgili motora doğrudan aktarılmasıyla da gerçekleştirilebilir.

4. DEĞERLENDİR

Bu aşamanın amacı, öğrencilerin öğrenme sürecindeki deneyimleri ve öğrendikleri üzerine düşünmelerini sağlamaktır. Bu sayede öğrenciler süreç içinde

deneyimlediklerini ve öğrendiklerini gözden geçirerek kendilerini değerlendirme ve tanıma fırsatı elde edeceklerdir. Öğrencilerden şu soruları yanıtlamaları istenebilir:

- Bugün yapılan etkinliklerden neler öğrendiniz?
- Verilen problemleri tanımlayınız (Problemi kendi cümleleri ile ifade etme).
- En çok hangi görevde zorlandınız? Bu zorlukların üstesinden nasıl geldiniz? (Problemin çözümü için hangi stratejileri kullandınız ve neden bu stratejileri seçtiniz?)
- Problemi çözerken ne gibi sıkıntılar yaşadınız ve bunların üstesinden gelmek için neler yaptınız?
- Kullandığınız yöntemler, bu sıkıntıları gidermekte başarılı oldu mu?
- Grup arkadaşınızla anlaşmazlığa düştüğünüz durumlar oldu mu ve bunların üstesinden gelmek için neler yaptınız?
- Grup arkadaşınızdan neler öğrendiniz?
- Dört kişilik grup çalışmasını değerlendiriniz? Olumlu ve olumsuz yönleri nelerdir?

Değerlendirme, öğrencileri sıkmadan, her soru için verilen cevaplar tatmin edici bir düzeye ulaşıncaya kadar devam ettirilir.

Proje Hazırlıyorum

Bu hafta, projelerin tamamlanması önerilmektedir. Projenin “geliştirme-test etme” süreci için planlamalar yapılmalıdır. Bütün süreçler tamamlandıktan sonra “sunum” süreci planlanmalıdır. Detaylar için EK'ler incelenebilir.

EK - Proje Hazırlıyorum

Öğrencilerin öğrendiklerini grup hâlinde uygulayabilecekleri proje geliştirme sürecinin, eğitimin yaklaşık son 4 haftasını kapsayacak şekilde planlanması beklenmektedir. Projelerin tanımlama, hazırlık, empati, fikir üretme, geliştirme-test etme ve sunum gibi etkinliklerle yapılması ve rehber öğretmenler gözetiminde öğrencilere ders dışı etkinliklerle destek olunması önerilmektedir.

1. HAZIRLIK (Projeye Başlama)

Etkinliğin Amacı: Gruplara açık uçlu bir proje vererek öğrencilerin yenilikçi fikir üretme, problem çözme, analitik düşünme ve grup hâlinde çalışma becerilerini geliştirmektir. Öğrenciler bu etkinlikte kendi robotlarını geliştirebildiklerini görürler ve bu sayede özgüvenleri artar.

Açıklama: Bu etkinlikten itibaren gruplar iki hafta boyunca projelerini tanımlama ve tasarlama üzerinde çalışacaklardır. Daha sonra, önceki haftalarda tanımlayıp tasarladıkları robotu oluşturmaları ve gerekli programları geliştirmeleri beklenir. Öğrenciler öncelikle çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üretirler. Öğrencilerin problem üzerinde sistematik bir şekilde çalışmalarını için grupların problemi analiz etmeleri ve kendi çözüm önerilerini tasarlamaları, geliştirmeleri ve değerlendirmeleri istenir. Bu süreçte karşılaşılabilecek tasarım ve programlamayla ilgili problemlerde rehber öğretmenler, öğrencilere uygun zaman ve miktarda destek olurlar. Bu destek öğrencilerin ihtiyacından ne az ne de fazla olmalıdır.

Proje çalışmasında son ürünü geliştirmek için "Tasarım ve Üretim" dersinde uygulanan tasarım odaklı düşünmenin beş aşamasının uygulanacağı söylenir.

1. Empati kurma: Hedef kullanıcı kitlenin ihtiyaç ve beklentilerini irdeleme.
2. Tanımlama: Kullanıcılardan alınan bilgilerden çıkan sonuçları yorumlayıp ihtiyacı öngörme.
3. Fikir üretme: Çözüm için çeşitli, yaratıcı ve yenilikçi fikirler üretme.
4. Prototip geliştirme: Fikirlerin hızlıca uygulandığı prototipler hazırlama.
5. Test etme: Ortaya çıkan ürünü yeniden hedef kitle ile değerlendirme.

Sonraki Haftaya Hazırlık: Bir sonraki hafta için öğrencilerin çevrelerinden ve günlük hayatlarından robotlar kullanılarak çözülebilecek bir problemi belirlemeleri gerektiği iletilir. Problemi belirlemeden önce "empati kurma" adımını gerçekleştirmeleri söylenir. Uygun bir problem sorusu belirlemeden önce problemi anlamak gerekir. Belirlenen problem, projenin zorluklarını ve hedeflerini tanımlamaya yardımcı olur. Sonrasında problemle ilgili yoğun bir araştırma ve gözlem yapılması gerekir. Süreç, projenin tamamlanması için gerekli bilgiler sağlar ve ortam hakkında bilgiler sunar. Süreç sonunda çok vakit ve emek harcanarak geliştirilen robotun, insanların işine yaraması, bir problemini çözmesi ve hayatlarını kolaylaştırması istenir. Empati süreci sonunda

tasarlanan robotun kullanılabilmesi için planlanan/öngörülen ortamlardaki ihtiyaçların ve beklentilerin iyi belirlenmesi gerekir.

2. EMPATİ

Etkinliğin Amacı: Öğrenciler çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üreteceklerdir.

Açıklama: Gruplardan belirledikleri robot projesi için yaptıkları empati çalışmasının sonucunu ve problem tanımlarını sınıfta açıklamaları istenir. Açıklamalarının aşağıdaki bilgileri içermesi gerektiği belirtilir. Empati çalışması sonucunu sunmaya başlamadan önce gruplara hazırlanma, gözden geçirme ve tekrar etme için 10 dakika süre verilir.

- Problem cümlesi: Robotun ne yapması isteniyor?
- Robotun kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?
- Robotun kullanılması planlanan ortamdaki beklentiler nelerdir?

Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.

Rehber Öğretmene Not: Öğrenci tarafından belirlenen robot görevinin gerçekleştirilebilir olması önemlidir. Atölyelerde bulunan imkânlarla bu görevin başarılabilir olması gerekir. Bu safhada gerçekleştirilemeyecek bir görevin seçilmesi veya öğrencilerin bu tür bir projeye yönlendirilmesi, proje sürecinde vakit kaybetmelerine ve motivasyonlarının olumsuz etkilenmesine neden olabilir. Bu nedenle rehber öğretmenin proje önerilerini çok iyi incelemesi/irdemesi ve proje önerilerinin gerçekleştirilebilir/yapılabilir olduğundan emin olması gerekir.

Sonraki Haftaya Hazırlık: Öğrencilerden, sonraki hafta için belirledikleri projeler için "tanımlama" adımlarını gerçekleştirmeleri istenir. Tanımlama adımı, "empati" adımı, toplanan bilgiler bir bütün hâline getirilir. Süreçte ortaya çıkabilecek problemleri çözebilmek için nelerin yapılması gerektiği ile ilgili fikirler ortaya konur. Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir.

3. TANIMLAMA

Etkinliğin Amacı: Öğrenciler çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üretirler.

Açıklama: Gruplardan belirledikleri robot projesi için yaptıkları empati ve tanımlama adımlarının sonuçlarını ve projelerini sınıfta açıklamaları istenir. Açıklamalarının aşağıdaki bilgileri içermesi gerektiği belirtilir. Empati ve tanımlama çalışmasının sonuçlarını sunmaya başlamadan önce gruplara hazırlanma, gözden geçirme ve tekrar etme için 10 dakika süre verilir.

- Problem cümlesi: Robotun ne yapması isteniyor?
- Robotun kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?
- Robotun kullanılması planlanan ortamdaki beklentiler nelerdir?
- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?

- Robotun belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?
- Robotun belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekmektedir?

Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.

Sonraki Haftaya Hazırlık: Sonraki hafta için belirlenen işlemlerin gerçekleştirilebilmesi için olası çözümlerin fikir üretme aşamasında belirlenmesi ve uygun çözümlerin seçilmesi gerekir. Fikir üretme aşamasında, empati ve tanımlama adımı toplanan bilgiler bir bütün hâline getirilir ve çözüm önerisi taslak olarak ortaya konur. Süreçte ortaya çıkabilecek problemleri çözebilmek için nelerin yapılması gerektiği ile ilgili fikirler sunulur. Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir.

Rehber Öğretmen İçin Tanım: Fikir üretme aşamasında çok sayıda fikir ve çözüm üretmek önemlidir. Bu çalışma "Beyin Fırtınası" tekniğiyle gerçekleştirilebilir. Ayrıca problem alanının genişletilerek olaylara dışarıdan bakılması önerilir. Böylece sorunu incelemek ve çözüm bulmak üzere farklı yöntemler aranmaya başlanır.

4. FİKİR ÜRETME

Etkinliğin Amacı: Öğrenciler çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üretirler.

Açıklama: Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir. Bu hafta, empati ve tanımlama adımı toplanan bilgiler bir bütün hâline getirilir ve çözüm önerisi taslak olarak ortaya konur. Farklı çözümlerin önerilmesi gereklidir. Bunlar çok detaylı ve tamamlanmış fikirler olmak zorunda değildir fakat öneriler gerçekçi ve üretilmeye uygun olmalıdır.

Gruplardan belirledikleri robot projesi için yaptıkları fikir üretme çalışmasının sonucunu sınıfta açıklamaları istenir. Açıklamaların aşağıdaki bilgileri içermesi gerektiği belirtilir. Bundan önce gruplara hazırlanma, gözden geçirme ve tekrar etme için 10 dakika süre verilir.

- Problem cümlesi: Robotun ne yapması isteniyor?
- Robotun kullanılması planlanan ortamdaki ihtiyaçlar nelerdir?
- Robotun kullanılması planlanan ortamdaki beklentiler nelerdir?
- Hedeflerin gerçekleştirilmesi için yapılması gerekenler nelerdir?
- Robotun belirlenen hedefleri gerçekleştirebilmesi için tasarımı nasıl olmalıdır?
- Robotun belirlenen hedefleri gerçekleştirebilmesi için hangi işlemleri yapabilmesi gerekir?
- Problem için üretilen alternatif tasarım ve programlama çözümleri nelerdir?
- Tasarım ve programlama çözümünün seçim süreci nasıl gerçekleştirilmiştir?
- Seçilen programlama çözümünün şematik veya maddeler hâlinde gösterimi (algoritması) nasıldır?

Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.

Sonraki Haftaya Hazırlık: Gruplara sonraki hafta, üretilen fikirlere bağlı olarak robotun tasarlanıp programlanması aşamasına geçileceği söylenir.

5. GELİŞTİRME – TEST ETME

Etkinliğin Amacı: Öğrenciler tanımladıkları problemi çözmek için kendi tasarladıkları bir robotu geliştirir ve test ederler.

Prototip Geliştirme: Grupların belirledikleri robot projesi için yaptıkları fikir üretme çalışması sonucunda üretilmesine karar verdikleri robotu oluşturup programa başlamaları gerekir.

Gruplardan, fikir üretme aşamasında belirledikleri çözümleri etkinlik sonuna kadar hızlıca uygulamaları, robot tasarımı ve programı geliştirmeleri istenir. Çözüm için geliştirilen robotlarda ve programlarda bazı aksaklıkların/problemlerin olması olağandır. Planlanan robotun ilk denemede oluşturulması ve geliştirilen programın sorunsuz çalışması çok karşılaşılan bir durum değildir. Bu tür sorunların ortaya çıkmasının olağan olduğu rehber öğretmen tarafından öğrencilere söylenmeli ve/veya hissettirilmelidir. Ortaya çıkan sorunların çözümüne yönelik bir plan hazırlanıp uygulanmalıdır. Çözüm için gruplara aşağıdakilere benzer bir süreç önerilebilir:

- i. Problemler maddeler hâlinde listelenmeli (beklenen robot davranışı ile gerçekleşen arasındaki farklar),
- ii. Problemler arasındaki ilişkiler belirlenmeli,
- iii. Öncelik sırası belirlenmeli,
- iv. Öncelik sırasına göre çözümler geliştirilmeli (Çözüm için hipotezler oluşturulmalı),
- v. Geliştirilen çözümler (hipotezler) denenmeli ve
- vi. Çözümlerin her koşulda istenen şekilde çalışması için gerekli düzenlemeler yapılmalıdır.

Bütün grupların, hazırladıkları prototipleri (tam olarak çalışmasa bile) rehber öğretmene göstermesi sağlanır. Böylece rehber öğretmenin, geliştirilen ürünü test etmesi/görmesi, çeşitli çözüm önerileri getirmesi ve/veya gözden kaçan noktaları ortaya çıkarması sağlanabilir.

Test Etme: Ortaya çıkan son ürünü hedef kitle ile/hedef ortamda değerlendirmek geliştirilen çözümlerin kusursuz bir şekilde çalışması için önemlidir. Hedef kitleye sınıfta ulaşmak her zaman mümkün olmayabilir. Bu nedenle grupların, ortaya çıkardıkları ürünleri diğer öğrencilerin kendi başlarına deneyimlemelerine izin vermesi, gözden kaçan unsurların ortaya çıkarılmasına yardımcı olabilir.

6. SUNUM

Projenin son aşamasında gruplara projelerini sunmaları konusunda yardımcı olunmalıdır. Gruplar projelerini bir senaryo çerçevesinde teatral olarak sunabilir.