

“СОНИРХОЛТОЙ АРДУИНО” Өөрсдөө зохион бүтээцгээе

Хоёрдугаар дэвтэр

Улаанбаатар хот
2025 он

ДАА – 370

НАА – 74

X - 84

ISBN: 978 9919 0 4517 3

Тус номонд техник, технологийг анхлан сонирхогч хүүхэд залуучуудад зориулан хялбар шийдэл, бүхий ухаалаг төхөөрөмж, роботыг зохион бүтээж, түүнийг програмчлан удирдах бодлого дасгалын хураамжуудийг түүвэрлэн оруулсан болно. Энэхүү номны тусламжтайгаар залуу зохион бүтээгчид Ардуино-микрокомпьютерийн үндсэн ойлголт, ажиллагаатай танилцах, Ардуино програмчлалын C хэлийг ашиглан математик, физикийн суурь ойлголт бүхий тооцооллуудыг хийх, мөн оролт гаралтын төхөөрөмжүүдийн ажиллагаатай танилцаж, тэдгээрийн бодит хэрэглээ болох гэрлэн, мэдрэгч, моторыг ашиглан хөдөлгөөнт дүрслэл, робот удирдлагын системийг симуляцийн орчинд загварчлан програмчлах, түүнийг Ардуино програмчлалын орчинд хөрвүүлэн ашиглах аргачлалд бие даан суралцана. Түүнээс гадна өөрсдийн зохион бүтээсэн роботоор ухаалах шийдэл бүхий үйлдлүүдийг гүйцэтгэх болно.

“СОНИРХОЛТОЙ АРДУИНО ”

Өөрсдөө зохион бүтээцгээе

Хоёрдугаар дэвтэр

**Зохиогч. Доктор (Ph.D) Ц.Хүрэлбаатар
Доктор (Ph.D) С.Баттуяа
Г.Дорж**

Хаяг: “ЛИОНРОБО” ХХК
Улаанбаатар хот, Сүхбаатар дүүрэг, 9-р хороо, 7-р хороолол
Баян бүрдийн тойрог 14182, 54-р барилга
Компьютер худалдааны төв, 4003 тоот

Э-шуудан: lionrobo77@gmail.com, uhaan.tbb111@gmail.com

Утас: +(976)-91999222, +(976)-91816895, +(976)-99175371

Цахим мэдээлэл: www.facebook.com/LionRobo2016

Редактор: Р.Батчимэг

ГАРЧИГ

ӨМНӨХ ҮГ

| | |
|---|----|
| БҮЛЭГ 1. ПРОГРАММ БИЧИХ АРГАД СУРАЛЦАЦГААЯ | |
| 1.1 Програм бичих аргад суралцацгаая | 5 |
| 1.2 Тоглонгоо програм бичиж сурцгаая | 6 |
| 1.3 Симуляцийн орчинд програм бичиж сурцгаая | 8 |
| 1.4 Симуляцийн орчин дахь хөдөлгөөнт роботын програмчлал | 13 |
| 1.4.1. Физикийн хичээлээр олж авсан мэдлэгээ роботын явах замыг тооцоолоход хэрхэн ашиглах вэ? | 15 |
| 1.4.2 Мэдээлэл зүй, Математикийн хичээлээр сурсан мэдлэгээ роботыг жолоодоход хэрхэн ашиглах вэ? | 17 |
| 1.4.3. Технологи, зургийн хичээлээр сурсан мэдлэгээ робот зохион бүтээхэд хэрхэн ашиглах вэ? | 22 |
| БҮЛЭГ 2. СИМУЛЯЦИЙН ОРЧНООС АРДИУНО ПРОГРАМЧЛАЛЫН ОРЧИНД | 24 |
| 2.1.Симуляцийн орчинд физик загварыг хэрхэн үүсгэх вэ? | 25 |
| 2.2. Мэдрэгч, түүний програмчлал | 31 |
| 2.3 Хар цагаан өнгө мэдрэгч, түүний програмчлал | 36 |
| БҮЛЭГ 3. АРДИУНО ХАВТАНГИЙН ПРОГРАМЧЛАЛ | 43 |
| 3.1.ArM2_V2 модуль интерфэйсийн гэрлэн эффект програмчлал | 45 |
| 3.2 ArMV2_Модуль V2-ийн оролт, гаралтын интерфэйсийг хослуулан хэрэглэх нь | 47 |
| 3.3 ArM1_V2 Модуль 1 ийн Серво моторын удирдлага | 51 |
| 3.4 Bluetooth модуль ашиглан Ардуиног удирдах (HC-05): | 52 |
| 3.5 Хэт авианы мэдрэгч (HC-SR04) ашиглан зайг хэмжих: | 54 |
| 3.6 Чийг мэдрэгч (FC-28) модулийг ашиглан хөрсний чийгшлийг тодорхойлох | 55 |
| БҮЛЭГ 4. УХААЛАГ РОБОТЫН ПРОГРАМЧЛАЛ | 58 |
| 4.1 Роботын загварчлал | 59 |
| 4.2 Моторын удирдлагын програмчлал | 60 |
| 4.3 Bluetooth модуль ашигласан алсын удирдлагатай роботын програмчлал | 62 |
| 4.4 Контактан мэдрэгч бүхий роботын програмчлал | 63 |
| 4.5 Эргэлт мэдрэгч бүхий роботын програмчлал | 66 |
| 4.6 Зураас дагагч роботын програмчлал | 67 |
| 4.7 Дасгал 4.7. Зураас дагагч роботын програмчлал | 76 |

ӨМНӨХ ҮГ

Хүний үйл ажиллгааг хөнгөвчлөх төхөөрөмжийг зохион бүтээх нь олон зууны турш хүн төрлөхтөний мөрөөдөл байсан. Гэтэл өнөөдөр энэхүү мөрөөдөл нь нэгэнт биелэлээ олж, ухаалаг төхөөрөмж, гар утас, компьютергүйгээр технологийн хөгжлийг төсөөлөхийг аргагүй болжээ. Технологийн дэвшлийн үрээр бидний амьдралд интерактив электрон тоног төхөөрөмжүүдийн хэрэглээ өсөж байгаа нь хүн төрлөхтөн шинжлэх ухаан технологийн хөгжлөөс илүү ихийг хүсэн хүлээдэг болсон байна.

Технологийн хөгжлийг өдөөгч гол хүчин зүйл нь хүний хязгаарлагдмал хүч, чадварыг орлохуйц тоног төхөөрөмж бүтээх хүсэл тэмүүлэл юм. Үүний нэгэн жишээ бол дэлхийн олон улсад оюутан сурагчид, хүүхэд багачуудын зохион бүтээх сэтгэлгээ, авьяас чадвар, техник сэтгэлгээг хөгжүүлэх, чөлөөт цагийг зөв боловсон өнгөрүүлэх, бие даан суралцах үйл ажиллагааг сайжруулах зорилготойгоор робот тоглоом, програмчлагддаг ухаалаг төхөөрөмжүүдийг сургалтанд ихээр хэрэглэж байгаа явдал юм.

Мөн үүнтэй уялдан роботын төрөл бүрийн уралдаан тэмцээнүүд зохион байгуулагддаг. Манай улсад энэ төрлийн уралдаан тэмцээнүүд (ABU Robocon, RobotSumo, Автомат робот, Мон робот, Робот хөл бөмбөг гэх мэт) цөөнгүй явагдах болсонтой уялдан оюутан сурагчид, хүүхэд багачууд өөрсдийн зохион бүтээсэн роботоор үндэсний болон олон улсын тэмцээнүүдэд оролцох шаардлага тавигддаг.

Сүүлийн үед манай улсад гадаадын сургалтын төхөөрөмж, роботуудаар (LEGO, VEX, бусад) тоноглогдсон сургалтууд цөөнгүй явагдах болсон. Гэвч тэдгээр нь эвдэрсэн тохиолдолд дотооддоо засварлах боломж хязгаарлагдмал, өртөг өндөртэй, сурагчдын хувьд төхөөрөмжийн ажлын заавар гадаад хэл дээр байх зэрэг хүндрэлтэй асуудлууд тулгардаг. Мөн технологийн хувьд цаашид хөгжүүлэх боломж нээлттэй биш, хязгаарлагдмал байдаг.

Монголчууд бид ч дэлхийтэй хөл нийлүүлэн, техник технологийн хөгжлийг даган, хэрэглэгч төдий байх бус өөрсдөө технологи хөгжүүлэгчид болох шаардлага зайлшгүй тулгарч байгаа билээ. Монгол ардын оньсон тоглоомоос өөр оюуны чадварыг дээшлүүлэх тоглоом, сургалтын төхөөрөмжүүд цөөн. Иймд үндсэний үйлдвэрлэлийн "Сургалтын робот" зохион бүтээхээр дараах зорилго, зорилтуудыг дэвшүүлж байна. "Монгол улсын боловсролын салбарт 2010 он хүртлэх хугацаанд Мэдээлэл, харилцаа холбооны технологийг нэвтрүүлэх үндсэн чиглэл"-ийг хэрэгжүүлэх асуудлын талаар хэлэлцэж зөвлөмж гаргасан. Уг зөвлөмжинд "Боловсролын салбарт мэдээлэл, харилцаа холбооны технологийг нэвтрүүлэх цаашдын зорилт"-ыг тодорхойлохдоо "2020 он гэхэд боловсролд мэдээллийн технологийг бүрэн нэвтрүүлнэ." хэмээн 4 зорилт бүхий үндсэн зорилгоо тодорхойлж байжээ. Дээрх зорилтууд дунд сургалтын талаар тусгахдаа "Мэдээлэл, харилцаа холбооны технологийн мэдлэг, чадвар эзэмшүүлэхэд чиглэгдсэн сургалтыг суралцагчдын хэрэгцээ, сонирхолд тулгуурлан зохион байгуулах уян хатан тогтолцоог бий болгох" хэмээн заасан байдаг.

Мөн БШУ-ны сайд 2012 онд хүүхдийн авьяасыг нээж хөгжүүлэх тухай тушаал гарган үндэсний хэмжээнд "Авьяас" хөтөлбөрийг хэрэгжүүлж байгаа. 2012-2013 оны хичээлийн жилд ЕБС-ийн нийт сурагчдын 20% нь дугуйлан, секцэд хамрагдан суралцаж байсан бол 2014-2015 оны хичээлийн жилд нийт сурагчдын 115% /нэг сурагч хоёроос гурван дугуйланд хамрагдаж байгааг тооцсон/ нь дугуйлан, секц, клубт хичээллэж авьяас чадвараа нээн хөгжүүлж байна. Энэ хугацаанд нисэх загвар, сэтгэх чадвар, бүтээлчээр уншигч, легоробот, гитарын клубүүдийг улсын хэмжээнд загвар байдлаар туршин хэрэгжүүлжээ.

МУ-ын боловсролын салбарт хэрэгжүүлж буй дээрх бодлогын баримтуудаас харахад МХХТ-ийг ЕБС-ийн сурагчдад олгож буй цар хүрээ нь нэмэгдэж буйтай холбогдуулан их, дээд сургуулийн боловсролын хөтөлбөрүүд ч мөн өөрчлөгдөж, сургалтын тоног төхөөрөмжүүд ч илүү ухаалаг шинэлэг болох хандлагатай нь харагдаж байна.

Монголын залуу инженерүүд дэлхийд танигдах болтугай.

Зохиогчоос

Бүлэг 1

Програм бичих аргад суралцацгаая



Бүх компьютеруудыг **техник** болон **програм хангамж** гэсэн хоёр хэсгээр бүтээдэг. **Техник хангамж** гэдэг нь компьютерын физик хэсгүүдийг хэлнэ. Жишээ нь: Роботын хувьд авч үзвэл гэрэл, удирдлагын төхөөрөмж, мэдрэгч, мотор зэрэг байж болно.

Програм хангамж гэдэг нь техник хангамжийг удирдах зааварчилгаа, код юм. Жишээ нь: роботын удирдлагын програм, командууд энд хамаарна.

Компьютеруудын техник хангамжийг програм хангамжтай холбож ажиллуулахад **програмын хэлийг** ашиглана.

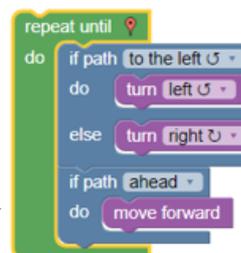
Програмын хэл гэдэг нь компьютерын програмыг бичихэд хэрэглэдэг тооцоолон бодох чадвар бүхий дүрэм болон командын бүрдэл юм.

Дэлхий дээр 700 орчим өргөн хэрэглэгддэг програмчлалын хэл байдаг.

Жишээлбэл:

- Python
- Blockly
- MATLAB
- Java
- Basic
- Arduino C
- Assembler

```
main.py
1 # Example: Printing even numbers from a list
2 numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
3 for num in numbers:
4     if num % 2 == 0:
5         print(num, "is even")
```



```
Blink | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
Blink.ino
1
2 // the setup function runs once when you press reset or power the board
3 void setup() {
4     // initialize digital pin LED_BUILTIN as an output.
5     pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 // the loop function runs over and over again forever
9 void loop() {
10    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltag
11    delay(1000); // wait for a second
12    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the volt
13    delay(1000); // wait for a second
14 }
```

```
>new
>10 print "HELLO"
>run
HELLO
OK
>
```

Тэдгээр програмчлалын зарим хэлүүдийн онцлогыг дурдвал:

- Ассемблер хэл нь компьютерын техник хангамжтай шууд холбогдоход зориулагдсан хамгийн доод түвшний програмчлалын хэл юм.
- Харин Basic хэл бол хэрэглэхэд хялбар байх ерөнхий зориулалттай зохион бүтээгдсэн дээд түвшний програмчлалын хэлнүүдийн нэг юм.
- Си програмчлалын хэл бол хамгийн эртний програмчлалын хэл бөгөөд сүүлийн үеийн дэвшилтэт програмчлалын хэл бүхэн Си хэлнээс эхтэй гэж үзэж болно. Үүний нэг жишээ бол бидний судлахаар төлөвлөж байгаа Ардуино болон блок програмчлалын хэлүүд юм.

Бид энэ удаагийн сургалтаараа нээлттэй сан бүхий 1. <https://blockly.games/maze?lang=en>, 2. <https://lab.open-roberta.org/>, 3. <https://www.arduino.cc/> сайтуудыг ашиглан тоглоом, симуляци, бодит орчнуудыг хослуулж, хэрхэн програмыг хялбар бичиж сурах аргыг заана.

1.2. Тоглонгоо програм бичиж сурцгаая

Компьютер ямар нэг ухаалаг үйлдэл хийж байна, тухайлбал шатар тоглож байна гэж төсөөлье. Одоо тэр үйлдлийг хүн тооцоолон гүйцэтгэж байна гэе. Хүн, компьютер хоёр хоёулаа заавар, дүрмийн дагуу ажилладаг. Гэвч хүн бол өөрийнхөө талаар бодож, шинэ мэдлэг дээр тулгуурлан аливаа зүйлд суралцаж, түүнийг өөрчилж чадна. Харин робот бол өөрийнхөөрөө сэтгэн бодож чадахгүй, зөвхөн хүний өгсөн зааврын дагуу л ажилладаг. Тэгэхээр програмын код бичих гэдэг нь тухайн програмын командын санг ашиглан логик дэс дараалалтай асуудлыг шийдвэрлэхийг хэлнэ.

Бүгдээрээ нээлттэй сан бүхий <https://blockly.games/maze?lang=en> сайтын Blockly Games тоглоомыг тоглонгоо програм бичих аргад суралцагсаая.

Blockly Games нь блок дээр суурилсан кодчилал ашиглан програмчлалын ойлголтуудыг заах зорилготой боловсролын цуврал тоглоом юм. Энэ нь Google-ээс боловсруулагдсан бөгөөд хүүхдүүд болон анхлан суралцагчдад програмчлалын үндсийг хөгжилтэй, интерактив хэлбэрээр танилцуулах зорилготой юм.

Энэ нь хэрхэн ажилладаг вэ:

Визуал кодчилал: Хэрэглэгчид эвлүүлдэг тоглоомтой адил код үүсгэхийн тулд блокуудыг чирж угсарна. Блок бүр нь давталт, нөхцөл, хувьсагч гэх мэт програмчлалын ойлголтыг илэрхийлдэг.

Текстэд суурилсан кодчилал байхгүй: Энэ нь блок дээр суурилсан тул програмын командыг бичих шаардлагагүй бөгөөд алдаа гарах магадлалыг бууруулж, суралцахад илүү ойлгомжтой болгодог.

Даалгавар сорилтууд: Тоглоомууд нь энгийнээр эхэлдэг боловч аажмаар төвөгтэй болж, давталт, нөхцөл, функц, логик зэрэг ойлголтуудыг агуулсан.

Тоглоомын төрлүүд:

Онъсого—Энгийн онъсого шийдвэрлэхийн тулд блокуудыг цэгцлэх замаар үндсэн ойлголтуудыг заана.

Maze—Төөрдөг байшинд шилжих замаар давталт болон логик нөхцөлийн ойлголтыг танилцуулна.

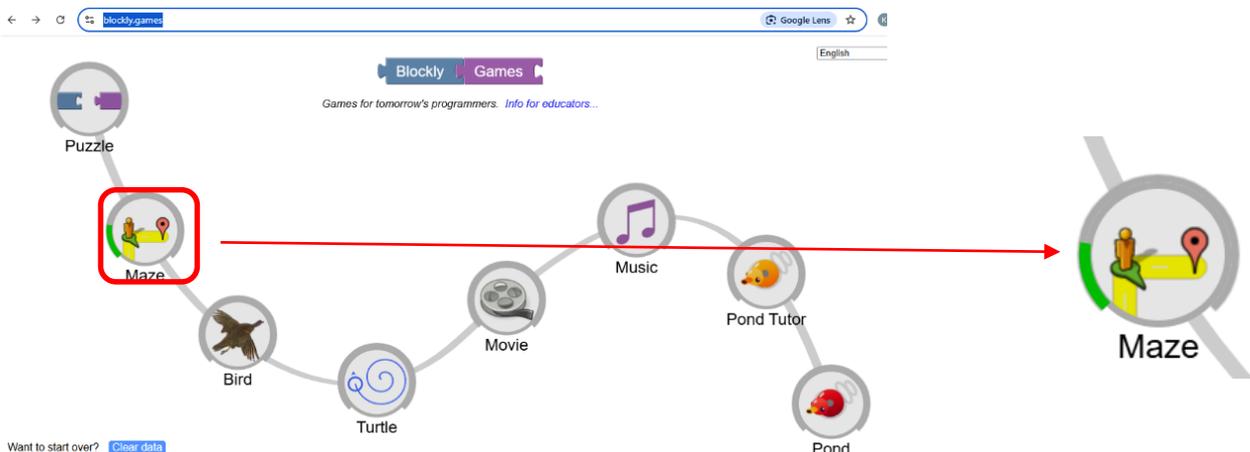
Шувуу - Шувууг зорилгодоо хүрэхэд нь туслах нөхцөл дээр анхаарлаа төвлөрүүлнэ.

Мэлхий— Зураг зурах давталт болон функцуудыг судална.

Кино – Хөдөлгөөнт дүрсийг зааврын дарааллаар харуулна.

Цөөрмийн багш - Цөөрөмд тулалдах замаар функцууд болон кодын үр дүнг заана.

Цөөрөм – Чөлөөт хэлбэрийн кодчилал тоглоом бөгөөд энэ нь асуудлыг нээлттэй шийдвэрлэх, стратеги боловсруулах боломжийг олгоно.



Maze буюу төөрдөг байшин тоглоомыг тоглонгоо програм бичих аргад суралцагсаа. Төөрдөг байшин тоглоом нь 1-10 : Maze 1 ●○○○○○○○○○(10) хүртлэх даалгавартай бөгөөд эхний түвшиний даалгаврыг дарааллын дагуу зөв угсарч ажиллуулсан суралцагч дараагийн шатруу орно.

1. Төөрдөг байшин даалгавар -1

Blockly Games : Maze 1 ●○○○○○○○○○(10)

Техник хангамж буюу гүйцэтгэлийн талбар

Блокын талбар

Програмын дараалал гэдэг нь алхам алхмаар, эмх цэгцтэйгээр ар араасаа гүйцэтгэгдэх үйлдлүүд юм.

Бичвэрийн талбар

Run Program

Даалгаврыг гүйцэтгэхийн тулд:

- Урагшаа
- Урагшаа гэсэн функцийг блокыг хоёр удаа дараалуулан байрлуулснаар үйлдлийг амжилтай гүйцэтгэнэ.

Гүйцэтгэлийн үр дүн

Блок програмын орчин

Congratulations!
You solved this level with 2 lines of JavaScript:
moveForward();
moveForward();
Are you ready for level 3?

Reset

2. Төөрдөг байшин даалгавар -2

Blockly Games : Maze 2 ●○○○○○○○○○(10)

Гүйцэтгэлийн үр дүн

Congratulations!
You solved this level with 5 lines of JavaScript:
moveForward();
turnLeft();
moveForward();
turnRight();
moveForward();
Are you ready for level 3?

Даалгаврыг гүйцэтгэхийн тулд:

- Урагшаа
- Зүүн эргэх
- Урагшаа
- Баруун эргэх функцийг блокуудыг дараалуулан байрлуулснаар үйлдлийг амжилтай гүйцэтгэнэ.

1.3. Симуляцийн орчинд програм бичиж сурцгаая

Энэ хэсэгт бүгдээрээ Германы боловсролын Ухаалаг шинжилгээ, мэдээллийн системийн институт (IAIS)-ын санаачилгаар хөгжүүлэгдсэн "Роберта – Роботтой суралцах нь" нэртэй програмчлалын орчинг ашиглаж програм бичих аргад суралцагсаа.

Нээлттэй Роберта лаборатори [/https://lab.open-roberta.org/](https://lab.open-roberta.org/) нь хэрэглэгчид, ялангуяа хүүхэд, өсвөр насныханд NEPO® хэмээх график програмчлалын хэлийг ашиглан төрөл бүрийн робот, микроконтроллеруудыг хялбар програмчлах боломжийг олгоход зориулагдсан.

Энэхүү платформ нь LEGO® MINDSTORMS® EV3, NXT роботууд болон Ардуино хавтанд суурилсан роботууд зэрэг төрөл бүрийн техник хангамжийн системийг дэмжин ажиллах боломжтой. Нээлттэй-Роберта лаборатори нь визуал програмчлалын аргыг ашигласнаар анхлан суралцагчдад зориулсан сургалтын үйл явцыг хялбаршуулж, програмын команд бүхий блокуудыг угсарч програмыг бичнэ.

Вэбд суурилагдсан энэхүү симуляцийн програм нь суулгах шаардлагагүй бөгөөд вэб хөтөч бүхий ямар ч төхөөрөмжөөс хандах боломжтой тул сургалтын танхимаас гадна гэрийн нөхцөлд онлайнаар суралцахад тохиромжтой. Энэхүү платформ нь англи, герман, франц, испани болон бусад хэлээр олон хэл дээр ашиглах боломжтой бөгөөд энэ нь дэлхий даяарх хэрэглэгчдийн хүртээмжийг өргөжүүлж байна.

Open Roberta Lab нь физик роботуудыг програмчлахаас гадна хэрэглэгчид өөрсдийн программыг виртуал робот дээр туршиж үзэх боломжтой симуляцийн орчныг санал болгодог. Энэ функц нь физик тоног төхөөрөмж ашиглахгүйгээр туршилт хийх, суралцах боломжийг олгодог.

Уг төслийг анх Google.org санхүүжүүлсэн бөгөөд програмчлалын боловсролыг илүү хүртээмжтэй, сонирхол татахуйц болгох, ялангуяа эмэгтэй оюутнуудыг STEM талбарт суралцахыг дэмжих зорилготой юм. Нээлттэй Роберта байгуулагдсан цагаасаа хойш 120 гаруй орны хэрэглэгчдэд хүрч, програмчлал, робот техникийн чиглэлээр дэлхийн боловсролын хүчин чармайлтад ихээхэн хувь нэмэр оруулж байна.

lab.open-roberta.org

robot help login tutorials gallery languages

Program your Robots & Boards

Welcome to Open Roberta Lab

Our open source platform »Open Roberta Lab« let you create your very own programs in no time at all using drag and drop.

Sign in

Log in and have access to your saved programs and settings. →

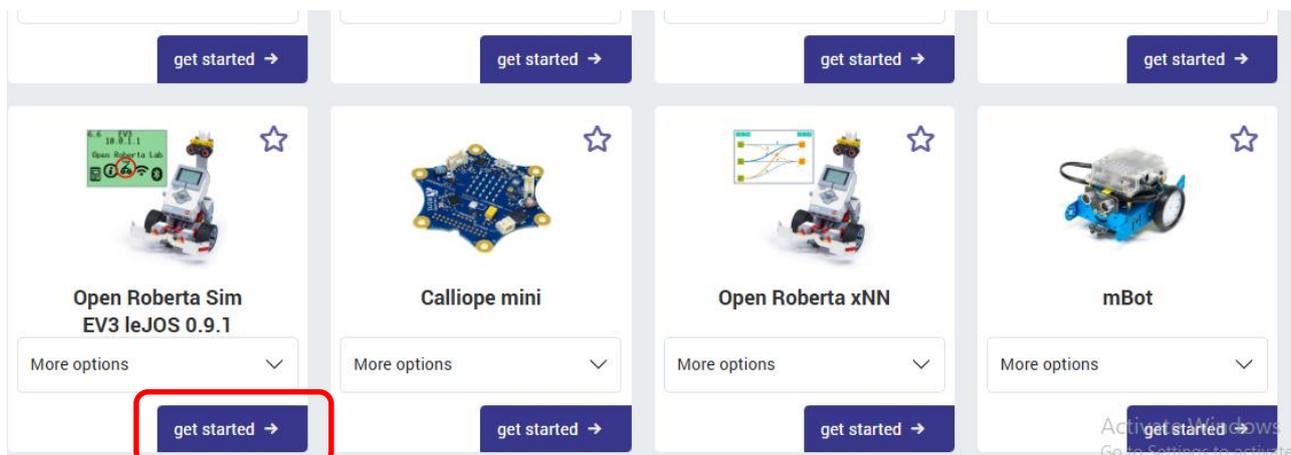
Import your NEPO program

Upload your XML file to our lab and continue where you left off. →

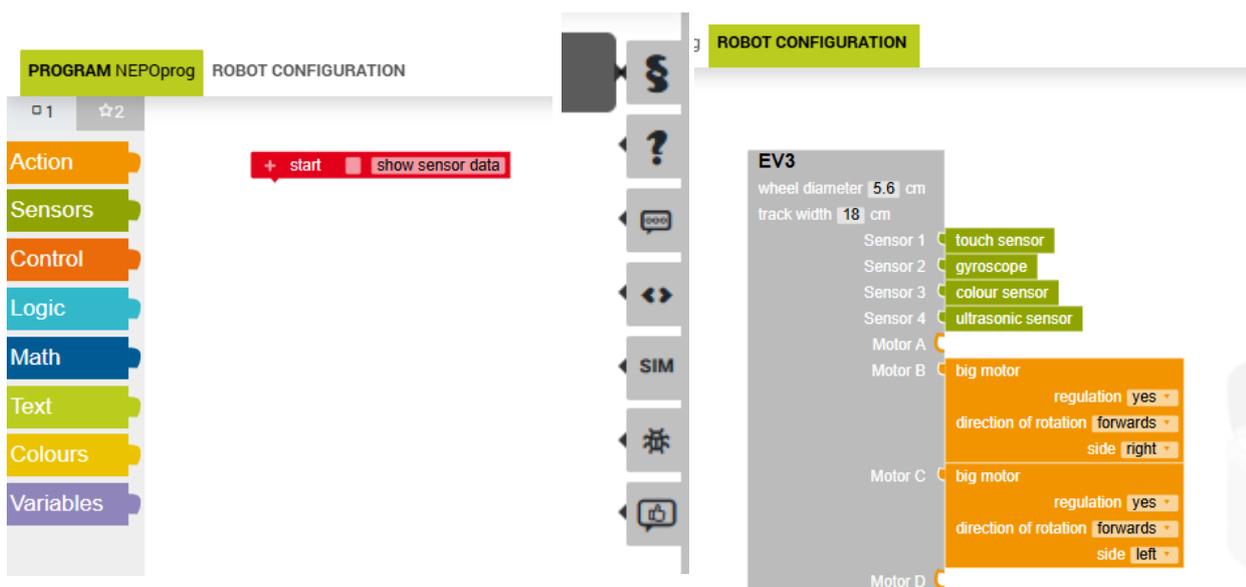
Take an interactive tour

Would you like to get started, but do not know exactly how? We will show you the first steps in an interactive tutorial. →

Бүгдээрээ нээлттэй-Роберта лабораторийн / <https://lab.open-roberta.org/> Open Roberta Sim E.V3 leJOS 0.9.1 симуляцийн талбарыг ашиглан програм бичигчээ.



Нээлттэй- Роберта лабораторийн Open Roberta Sim E.V3 ийн талбар нь *блок програмчлалын PROGRAM NEPOprog* талбар болон *оролт, гаралтын төхөөрөмжийн ROBOTCONFIGURATION* талбар гэсэн хоёр хэсгээс бүтнэ.



1.3.1 Симуляцийн орчинд загварыг хэрхэн хялбар зохион бүтээх вэ?

Бид “Open Roberta Lab” симуляцийн орчныг ашиглан оролт гаралтын төхөөрөмжүүдийг програмчлах, өөрсдийн программыг виртуал робот дээр туршиж үзэх зэргээр физик тоног төхөөрөмж ашиглахгүйгээр хэрхэн туршилт хийх, суралцаж болохыг мэдэж авлаа.

Иймд зохион бүтээгчид эхлээд “Ямар зориулалттай робот зохион бүтээх вэ? гэсэн асуултыг тодорхойлон дараах алхмуудыг хийнэ. Үүнд:

- Роботын гүйцэтгэх үйлдэл, даалгаврыг тодорхойлох
- Өгөгдсөн даалгаварт тохируулан роботын удирдлагын төхөөрөмжинд мэдрэгч, моторын холболтыг хийж загварчлах
- Өгөгдсөн даалгаврын дагуу явах роботын талбарыг бэлтгэх
- Програмыг хадгалах болон хадгалсан програмыг дахин ашиглах
- Робот удирдлагын төхөөрөмжийг програмчлах

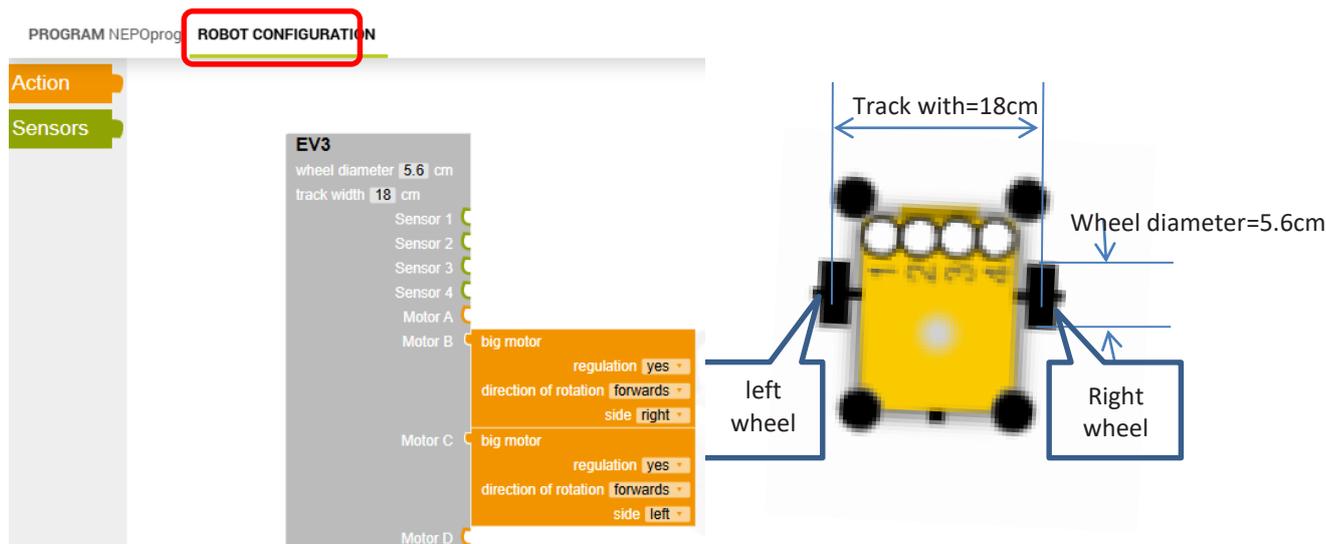
A. Роботын гүйцэтгэх үйлдэл, даалгавартай танилцах:

Даалгавар 1.3. “Open Roberta Lab” симуляцийн орчинд хоёр дугуйтай роботын дугуйны радиусыг 5.6 см, дугуй хоорондыг зайг 18 см байхаар роботыг загварчлан, роботыг эхлэх цэгээс 30 см урагш явахаар тооцоолон програмын кодыг бич.

B. Өгөгдсөн даалгаварт тохируулан роботын удирдлагын төхөөрөмжид мэдрэгч, моторын холболтыг хийж загварчлах:

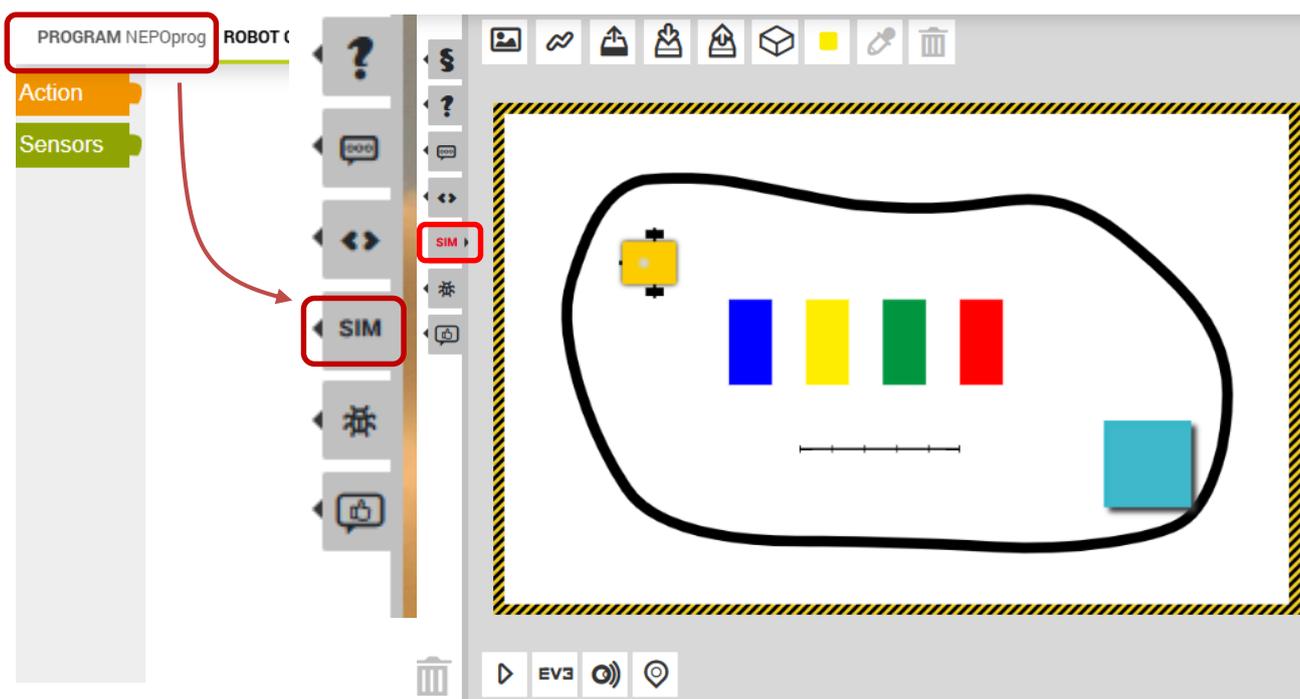
Open Roberta Sim E.V3 ийн оролт, гаралтын төхөөрөмжийн ROBOT CONFIGURATION талбарыг сонгож роботын мотор, дугуйнуудын холболтыг хийцгээе.

- Роботын дугуйны радиусыг 5.6 см
- Хоёр дугуйны хоорондыг зайг 18 см болгон тохируул



C. Өгөгдсөн даалгаврын дагуу явах роботын талбарыг бэлтгэх

Open Roberta Sim E.V3 ийн PROGRAM NEPOprog талбар дахь SIM товчлуурыг дарж роботын туршилтын талбарыг сонгоно.



D. Програмыг хадгалах болон хадгалсан програмыг дахин ашиглах

1. Програмыг хэрхэн хадгалах вэ?

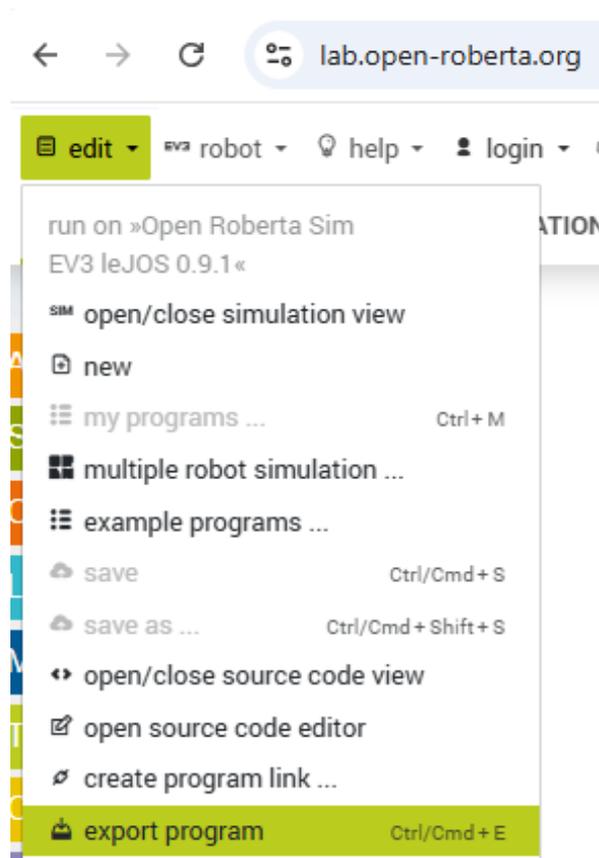
Алхам 1: Програм бичих

Open Roberta Lab (<https://lab.open-roberta.org>)-д нэвтэрч, программынхаа кодыг блок хэлбэрээр зохион байгуул.

Алхам 2: Файлыг хадгалах

1. Цэсийн дээд хэсэгт байрлах **"File" (Файл)** хэсгийг нээ.
2. **"Download" (Татаж авах)** сонголтыг дар.
3. Програм чинь таны компьютерт **.xml** өргөтгөлтэй файл хэлбэрээр татагдана.
4. **Ойлгомжтой нэр өгөх:**

Жишээ нь, "line-follower.xml" (зураас дагагч робот) гэж хадгалах



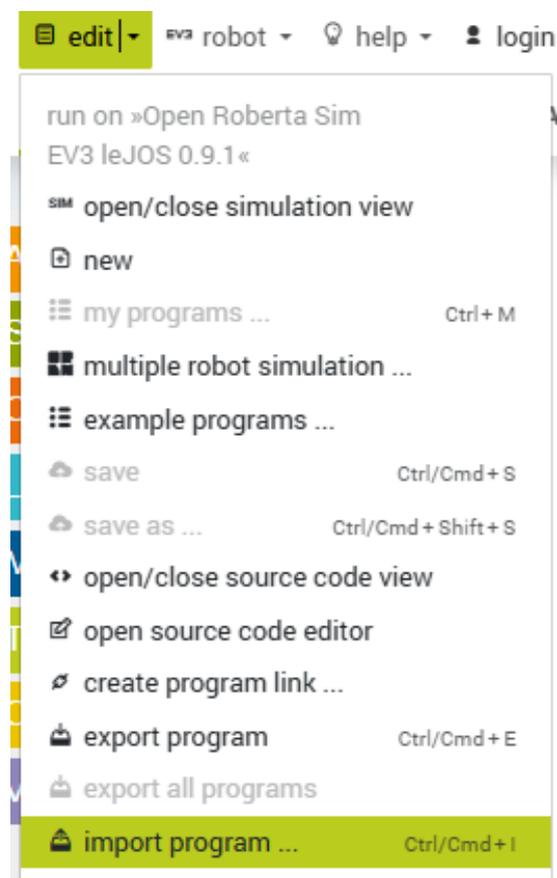
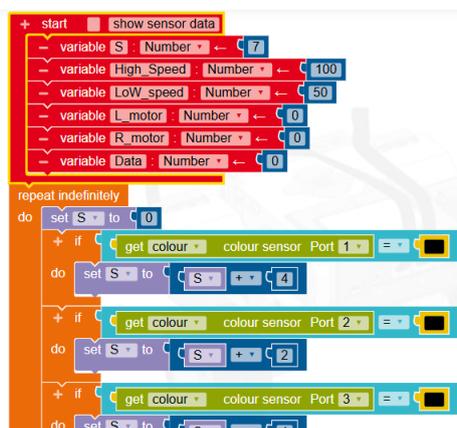
Хадгалсан програмыг дахин ашиглах

Алхам 1: Open Roberta нээх

- Open Roberta Lab руу орж, симулятор болон программчлалын орчныг ачааллана.

Алхам 2: Хадгалсан файлыг ачаалах

1. **"Edit" (Файл)** цэсийг нээ.
2. **"export program" (Файл оруулах)** товчийг дар.
3. **Өмнө хадгалсан .xml файлаа сонгоно.**
4. Программ автоматаар ачааллагдаж, таны ажиллаж байсан блок схем гарч ирнэ



Е. Робот удирдлагын төхөөрөмжийг програмчлах

Роботыг эхлэх цэгээс 30 см урагш яваад зогсохоор тооцоолон програмын кодыг бич.

1. Роботын моторыг хурдыг эхэлж сонгоод, явах хугацааг өөрчлөн тохируулах
2. Роботын явах хугацааг эхэлж сонгоод, роботын хурдыг өөрчлөн тохируулах
3. Роботын хурд болон явах хугацааг хоёуланд нь өөрчлөн тохируулах

Даалгавар 1.3.1. Роботын моторын хурдыг эхэлж сонгоод, явах хугацааг өөрчлөн тохируулах програмын кодыг бич.

Гүйцэтгэл 1.3.1. Өгөгдлийн дагуу даалгаврыг гүйцэтгэхийн тулд моторын хамгийн дээд хурд буюу 100% гэсэн сонголтыг хийж доорх дарааллын дагуу үйлдлийг гүйцэтгээрэй.

Даалгавар 1.3.2. Роботын явах хугацааг эхэлж сонгоод, роботын хурдыг өөрчлөн тохируулах програмын кодыг бич.

Гүйцэтгэл 1.3.2. Өгөгдлийн дагуу даалгаврыг гүйцэтгэхийн тулд роботын явах хугацааг 1500 милсекунд байхаар сонгоод, роботын хурдыг өөрчлөн тохируулаарай.

Даалгавар 1.3.3. Роботын явах хугацааг 600 милсекундээс их, роботын хурдыг 70 хувиас бага байхаар тохируулан програмын кодыг бич.

Гүйцэтгэл 1.3.3.

1.4. Симуляцийн орчин дахь хөдөлгөөнт роботын програмчлал

Симуляцийн орчин дахь хөдөлгөөнт роботын програмчлал нь бодит ертөнцөд роботыг бүтээхээс өмнө роботын үйл ажиллагааг загварчлах, турших, хөгжүүлэх боломжийг олгодог. Энэ нь роботын алгоритмуудыг аюулгүй, үр ашигтайгаар туршихад чухал үүрэг гүйцэтгэдэг.

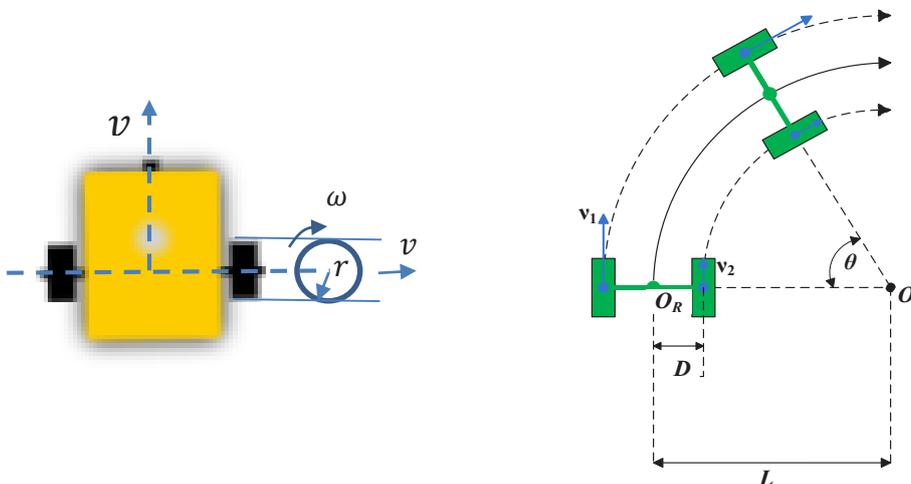
1.4.1. Физикийн хичээлээр олж авсан мэдлэгээ роботын явах замыг тооцоолоход хэрхэн ашиглах вэ?

Роботын удирдлагын төхөөрөмжийг зохион бүтээж, загварчлахад биетийн кинематик тооцооллыг судлах зайлшгүй шаардлага тавигддаг. Роботын кинематик нь тухайн биетийн хүчийг хянахгүйгээр түүний боломжит хөдөлгөөний хязгаарыг тодорхойлох процесс бөгөөд механикийн үндсэн хууль дээр тулгуурлан роботын бүтцэд тохируулж, тооцооллыг гүйцэтгэдэг.

Энэ хэсэгт бид энгийн хөдөлгөөнт роботын төрөл болох хөтлөгч бүхий хоёр үндсэн дугуйтай, дифференциал удирдлагатай хөдөлгөөнт роботын тооцооллыг авч үзье. Дифференциал удирдлагатай робот нь хоорондоо үл хамаарах жолоодлого бүхий хоёр дугуйгаар тоноглогдсон. Роботын байрлал нь X , Y координатын тэнхлэг дээрх проекцууд болон хөдөлгөөний чиглэлийн φ өнцгөөр тодорхойлогдох бөгөөд явах чиглэл нь хоёр дугуйны төвийг дайрсан тэнхлэгтэй перпендикуляр байна (зураг 1.4.1). Үүнээс гадна дугуйг гулгалтгүйгээр эргэнэ гэж үзвэл дугуйны шугаман хурдыг дараах хэлбэрээр илэрхийлж болно.

$$v = \omega r$$

Энд r - дугуйн радиус, v - дугуйны шугаман хурд, ω - дугуйны өнцөг хурд.

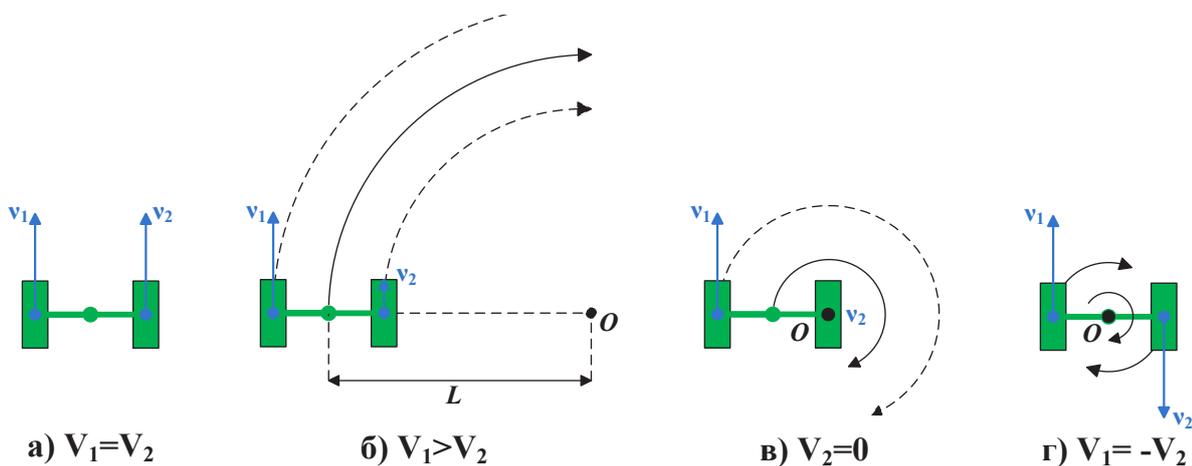


Зураг 1.4.1 Хөдөлгөөнт роботын кинематикийн дүрслэл

Дифференциал удирдлагатай роботын явах эд анги бүхий хоёр дугуйны тэнхлэг дээр роботын тэнхлэгийн төвийн цэг (O_R) байрласан гэж үзвэл тухайн хугацааны эгшин дэх роботын хурдны өөрчлөлтөөр тодорхойлогдох роботын эргэлтийн радиус ба эргэлтийн төвийн өнцөг хурд $\omega = \dot{\varphi}$ байна.

Энд: φ - нь роботын эргэлтийн өнцөг юм.

Хоёр дугуйны хурдны өөрчлөлтөөр тодорхойлогдох дифференциал удирдлагатай хөдөлгөөнт роботын замчлал, түүний дүрслэлийг зураг 1.4.2-д харууллаа



Зураг 1.4.2. Дифференциал удирдлагатай хөдөлгөөнт роботын эргэлтийн нөхцөлүүд

- Зүүн ,баруун дугуй ижил хурдтай үед
- Зүүн, дугуйны хурд баруун дугуйны хурднаас их үед
- Баруун дугуйны хурд "0" үед
- Зүүн, баруун дугуйны хурднууд эсрэг чиглэлтэй үед

Роботын төвөөс роботын эргэлтийн төв хүртэлх зайг L гэж тэмдэглэвэл баруун болон зүүн дугуйны хурдыг дараах хэлбэрээр илэрхийлж болно (Зураг 1.4.2.б).

$$V_L = \omega(L - D) \quad (1.4.2)$$

$$V_R = \omega(L + D) \quad (1.4.3)$$

Энд: D -нь роботын төвөөс дугуй хүртлэх зай.

Роботын шугаман хурд нь

$$V = \omega L \quad (1.4.4)$$

гэж тодорхойлогдох бөгөөд (1.4.2), (1.4.3) томъёог (1.5.4)-т орлуулбал

$$V = \frac{V_L + V_R}{2} \quad (1.4.5)$$

$$\omega = -\frac{V_L - V_R}{2R} \quad (1.4.6)$$

Эндээс роботын шугаман хурдыг X , Y тэнхлэгүүд дээр проекц $V_x = V \cos \varphi$, $V_y = V \sin \varphi$ гэж орлуулбал t_1 хугацааны эгшин дэхь роботын байрлал болон чиглэлийг дараах хэлбэрээр илэрхийлж болно.

$$\Delta x = X(t_1) - X_0 = \int_{t_0}^{t_1} V_x(t) dt = \int_{t_0}^{t_1} \frac{V_L(t) + V_R(t)}{2} \cos \varphi(t) dt \quad (1.4.7)$$

$$\Delta y = Y(t_1) - Y_0 = \int_{t_0}^{t_1} V_y(t) dt = \int_{t_0}^{t_1} \frac{V_L(t) + V_R(t)}{2} \sin \varphi(t) dt \quad (1.4.8)$$

$$\Delta \varphi = \varphi(t_1) - \varphi_0 = \int_{t_0}^{t_1} \omega(t) dt = -\int_{t_0}^{t_1} \frac{V_L(t) - V_R(t)}{2R} dt \quad (1.4.9)$$

Энд:

X_0, Y_0 - роботын анхны байрлалыг тодорхойлох координатын хавтгай дээр буулгасан проекц

φ_0 - робот хөдөлж эхлэх үеийн чиглэлийн өнцөг

Хэрэв зүүн, баруун дугуйны шугаман хурд $V_L(t)$, $V_R(t)$ нь мэдэгдэж байвал хугацааны дурын t агшин дахь роботын байрлал ба чиглэлийг тодорхойлох боломжтой гэдэг нь дээрх тэгшитгэлээс тодорхой харагдаж байна.

Дээрх тодорхойлолтоос харахад бидний сурсан математик болон физикийн суурь мэдлэг ямар чухал болох нь харагдаж байна.

Одоо бүгдээрээ микрокомпьютерийн програм хангамжийн тусламжтайгаар геометр болон хөдөлгөөний кинематик тооцоолол дээр тулгуурлан роботын явах замыг тодорхойлох хялбар туршилтийг хийж үзэцгээе.

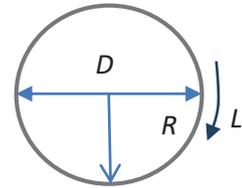
Даалгавар 1.4.1 Тойргийн радиус $r=5$ см, $\pi=3.14$ бол тойргийн урт L болон диаметр d –г олох програмын кодыг бич.

```

+ start show sensor data
- variable R_radius : Number ← 5
- variable Pi : Number ← 3.14
- variable d_diameter : Number ← 0
- variable L_length : Number ← 0

repeat indefinitely
do
set d_diameter to 2 × R_radius
set L_length to d_diameter × Pi
    
```

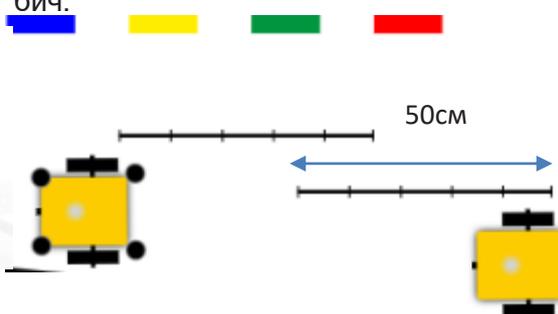
| - Variable Values | |
|-------------------|------|
| R_radius : | 5 |
| Pi : | 3.14 |
| d_diameter : | 10 |
| L_length : | 31.4 |



Даалгавар 1.4.2 Роботын зүүн болон баруун дугуйны явах хурдыг 30 хувиар, явах замын уртыг 50 см-ээр тохируулан програмын кодыг бич.

```

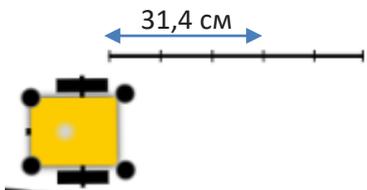
+ start show sensor data
steer forwards speed % left 30
speed % right 30
distance cm 50
    
```



Даалгавар 1.4.3 Роботын дугуйны тойргийн радиус $r=5$ см, $\pi=3.14$ бол 1 секундын хугацаанд робот ойролцоогоор 31,4 см шулуун замыг туулахын тулд моторын эргэлтийн хурдыг хэдээр сонгох вэ? Програмын кодыг бичиж туршилтын үр дүнгээр роботын моторын эргэлтийн хурдыг тохируулаарай.

```

+ start show sensor data
steer forwards speed % left 65
speed % right 65
wait ms 1000
    
```



Даалгавар 1.4.4 Роботын дугуйны тойргийн радиус $r=3.3$ см бол робот нь эхлэх байрлалаас 42 см орчим яваад зогсох програмын кодыг бич.

EV3

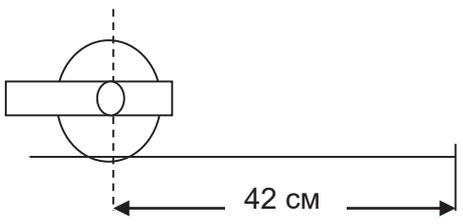
wheel diameter 6.6 cm

track width 18 cm

Sensor 1

Sensor 2

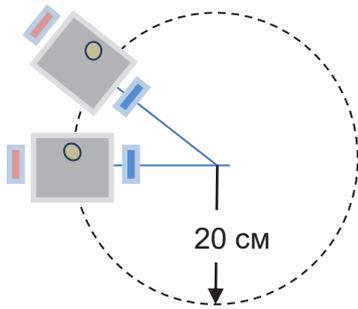
Бодолт: Дугуйны радиус $r=3.3$ см гэвэл дугуйны тойргийн урт $L=2\pi r=2*3.14*3.3=20.724$ болно. Хэрвээ бид роботын зүүн болон баруун моторын эргэлтийг ижил хурдаар 0.5 секундийн туршид 1 удаа бүтэн эргүүлэхээр тохируулан програмчилбал робот нь тооцооллын дагуу 1 секундын дараа 41,488 см буюу ойролцоогоор 42 см яваад зогсох болно.



```

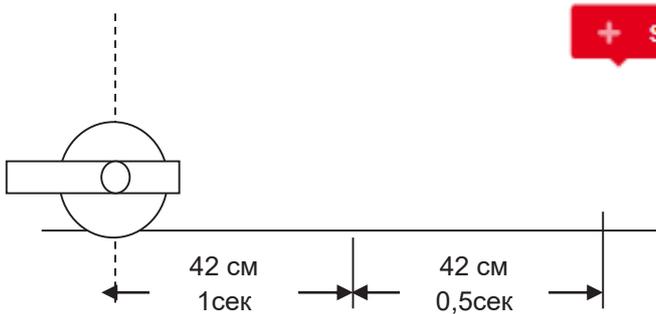
+ start show sensor data
steer forwards speed % left 94
speed % right 94
wait ms 500
wait ms 500
    
```

Даалгавар 1.4.5 Роботын зүүн, баруун дугуйны хурдыг тохируулан 20 см радиустай тойрог үүсгэн эргэлдэж явах роботын програмын кодыг бич.



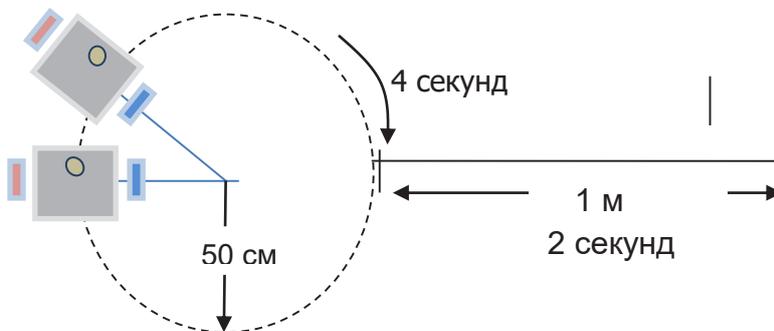
```
+ start  show sensor data
```

Даалгавар 1.4.6 Роботын дугуйны тойргын радиус $r=3.3$ см бол робот нь эхний 42 см замыг 1 секундэд явах ба дараагийн 42 см замыг 0.5 секундэд явахаар тооцоолон програмын кодыг бич.



```
+ start  show sensor data
```

Даалгавар 1.4.7 Роботыг эхний 4 секундэд 50 см-ийн радиустай нэг бүтэн тойрог үүсгэн эргэлдэх ба дараагийн 2 секундэд 1 метр шулуун явуулахаар тооцоолон програмын кодыг бич.



1.4.2. Мэдээлэл зүй, Математикийн хичээлээр сурсан мэдлэгээ роботыг жолоодоход хэрхэн ашиглах вэ?

Хүний үйл ажиллагааг хөнгөвчлөх ухаалаг төхөөрөмжүүд нь байгалийн физик шинж чанар дээр үндэслэгдэн математик тооцоолол бүхий програм хангамжийн тусламжтайгаар автоматаар унтрах, асах, мэдээлэл өгөх зэрэг олон үйлдэл гүйцэтгэх чадвартайгаар зохион бүтээгдсэн байдаг. (Жишээлбэл: Бидний ахуйн хэрэглээнд өдөр бүр ашиглагддаг будаа агшаагч, индүү, тоос соруулагч гэх мэт олон төхөөрөмжийг нэрлэж болно.)

Бид програм хангамжийн тусламжтайгаар энгийн математик тооцоолол ашиглан ухаалаг роботын явах замыг тодорхойлохын тулд **хоёрт** болон **аравтын** тооллын системийн ойлголтыг мэдэх шаардлагатай.

Хоёртын тооллын систем нь компьютерийн тооцоололд 0 ба 1 гэсэн хоёр цифрийг ашиглан тоон утгуудыг дүрсэлдэг тооллын систем юм. Энэхүү системийг логик гэйтүүдийн тусламжтайгаар тоон электроникийн хэлхээнд хялбараар хэрэгжүүлэх боломжтой тул орчин үеийн бүх компьютерийн дотоод бүтцэд хэрэглэгддэг.

Аравтын тооллын систем нь 0-9 хүртэлх 10 ширхэг цифрээс бүтэх орчин үеийн соёл иргэншлүүдэд хамгийн түгээмэл хэрэглэгддэг тооллын систем юм.

Програм бичихийн өмнө зоосон мөнгө орхиж нэгэн тоглоом тоглоцгооё.

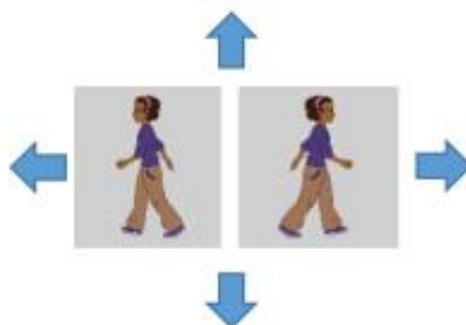
Нэг зоосон мөнгө орхиж **ХОЁР** өөр төрлийн үйлдэл хийж болно

| Сүлд | Тоо | Хийх үйлдэл |
|---|-----|---------------------------|
| 0 | 1 | $2^1=2$ |
|  | | Сүлд буудал ЗҮҮН тийш яв |
| | | Тоо буувал БАРУУН тийш яв |



Хоёр зоосон мөнгө орхиж **ДӨРВӨН** өөр төрлийн үйлдэл хийж болно

| 2^2 | 2^1 | Хийх үйлдэл $2^2=4$ |
|-------|-------|---------------------|
| 0 | 0 | УРАГШАА яв |
| 0 | 1 | ХОЙШОО яв |
| 1 | 0 | ЗҮҮН тийш яв |
| 1 | 1 | БАРУУН тийш яв |



Гурван зоосон мөнгө орхиж **ХЭДЭН** өөр төрлийн үйлдэл хийж болох вэ?

Аль нь зөв вэ?

- A. 6 B. 8 C. 9 D. 16

Хоёртын тооллын системээс Аравтын тооллын системд хөрвүүлэх хүснэгт нь тухайн 2 тын тооны тоо бүрийг 10 тын тооны эквивалент болгож хөрвүүлсэн хүснэгт юм. Доор 2 тын тооллын системээс 10 тын тооллын системд хөрвүүлэх жишээ хүснэгтийг үзүүлье:

| 2 тын тоо | 10 тын тоо |
|-----------|------------|
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 0101 | 5 |
| 0110 | 6 |
| 0111 | 7 |
| 1000 | 8 |
| 1001 | 9 |
| 1010 | 10 |
| 1011 | 11 |
| 1100 | 12 |
| 1101 | 13 |
| 1110 | 14 |
| 1111 | 15 |

Жишээ нь: 2 тын **1010** тоо нь 10 тын **10** тоонд хөрвөнө.

Хоёртын **1101** тоо нь аравтын **13** тоонд хөрвөнө.

1. 1101 тооны байрлалд аравтын утга тодорхойлно:

- $1 \times 2^3 = 8$
- $1 \times 2^2 = 4$
- $0 \times 2^1 = 0$
- $1 \times 2^0 = 1$



2. Нийт утга: $8 + 4 + 0 + 1 = 13$ (Аравтын тоо)

Хоёртын (бүтэн) тооллын системийн тоон утгыг аравтын (децимал) тооллын систем рүү хөрвүүлэх үед, эдгээр тоонууд нь роботуудын гүйцэтгэх үйлдлүүдийн логик болон механизмыг удирдах нөхцөлүүдийг тооцоолох зорилготой байдаг. Тэгэхээр, хоёртын системийн тооны хөрвүүлэлтийг хийх явцад гүйцэтгэх ёстой үйлдлүүдтэй танилцацгаая.

Роботын алгоритм эсвэл программчлалын нөхцөлүүдийг тооцоолох явцад, хоёртын тооны хөрвүүлэлт нь роботоос хэд хэдэн үйлдэл гүйцэтгэхийг шаарддаг. Жишээ нь:

1. **Роботын хөдөлгөөний удирдлага:**

- Хоёртын тоо нь роботын хөдөлгөөний чиглэл, хурд, болон зогсолтын нөхцөлүүдийг тодорхойлох боломжтой. Жишээлбэл: хоёртын тооны утга **0001** байвал робот нь зүүн тийш хазайлт үүссэнийг тодорхойлж, дараагийн шатанд хүрэхэд шаардлагатай нөхцөлийг тодорхойлж болно.

2. **Мэдрэгчийн өгөгдлийг боловсруулах:**

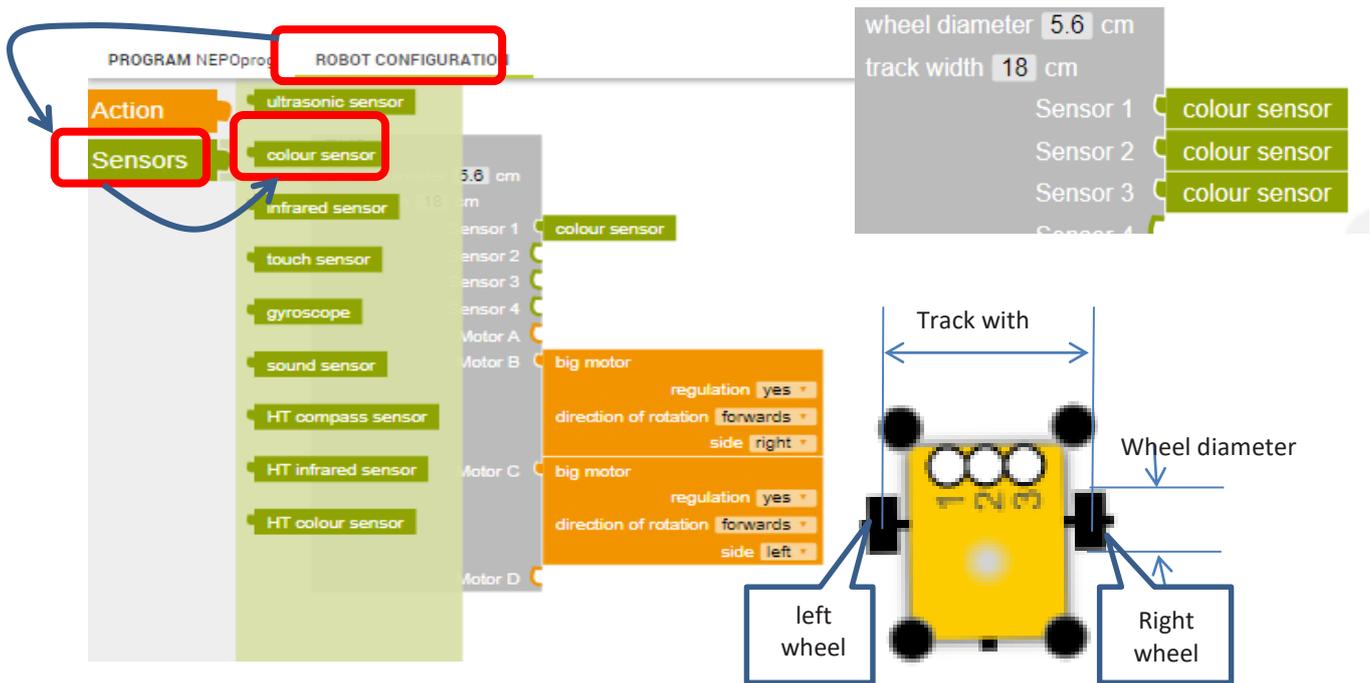
- Хоёртын тоог аравтын систем рүү хөрвүүлснээр робот мэдрэгчийн өгөгдлийг хурдан боловсруулах чадвартай болж, орчныг илүү үр ашигтай гүйцэтгэж чадна. Жишээ нь, хоёртын тоо **0110** нь роботоос орчны саадыг мэдрэх үед хэрхэн урьдчилан таамаглах алгоритм ажиллахыг тодорхойлно.

3. **Роботын амжилттай эсвэл алдаатай гүйцэтгэх нөхцөл:**

- Хоёртын тоо нь орчны нөхцөл байдлын үр дүнг гаргаж, аравтын систем рүү хөрвүүлсэн тоо нь үүнийг роботын гүйцэтгэлтэй холбож, зөвшөөрөгдөх нөхцөлүүдийг тогтооно. Хэрэв роботын програмчлалд **1111** гэсэн тоо ороход, энэ нь тодорхой нэг үйлдлийг гүйцэтгэх эсвэл зогсох шаардлагатай гэсэн үг болж болно.

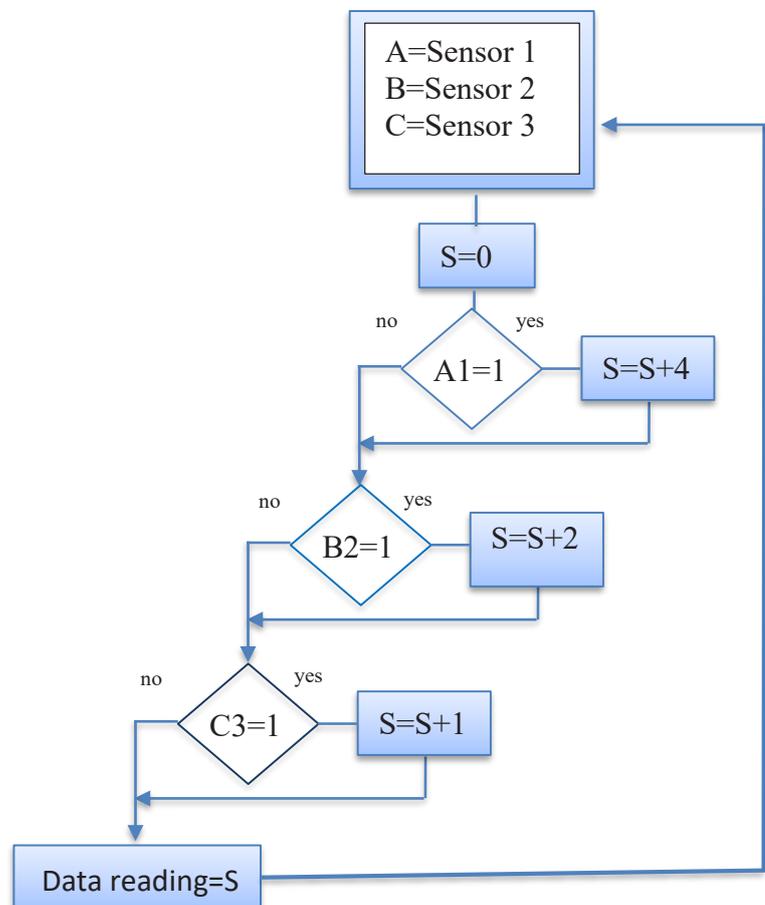
Хоёртын системийн тоог аравтын тооллын систем рүү хөрвүүлэх нь роботын програмчлалд үйлдлүүдийн шийдвэрлэх механизм, алгоритм боловсруулахад чухал үүрэг гүйцэтгэдэг. Энэ хөрвүүлэлтийн тусламжтайгаар робот нь зөв өгөгдөл боловсруулж, аюулгүй, зөв үйлдлүүдийг гүйцэтгэж чадна.

Open Roberta Sim E.V3 ийн ROBOTCONFIGURATION талбарт удирдлагын төхөөрөмжийн гурван ширхэг өнгө мэдрэгч, хоёр ширхэг моторын холболтыг хийцгээе.



Хоётын тооллын системийн гурван битийн тоог аравтын тооллын системд хөрвүүлсэн хүснэгт болон програмын алгоритмыг доор харуулсан байна.

| Хоёртын тоолол | Аравтын тоолол |
|----------------|----------------|
| A1 B2 C3 | S=Data reading |
| 0 0 0 | 0 |
| 0 0 1 | 1 |
| 0 1 0 | 2 |
| 0 1 1 | 3 |
| 1 0 0 | 4 |
| 1 0 1 | 5 |
| 1 1 0 | 6 |
| 1 1 1 | 7 |



Мөн дээрх програмын алгоритмд тохируулан бичсэн програмын кодыг бичицгээе.

```

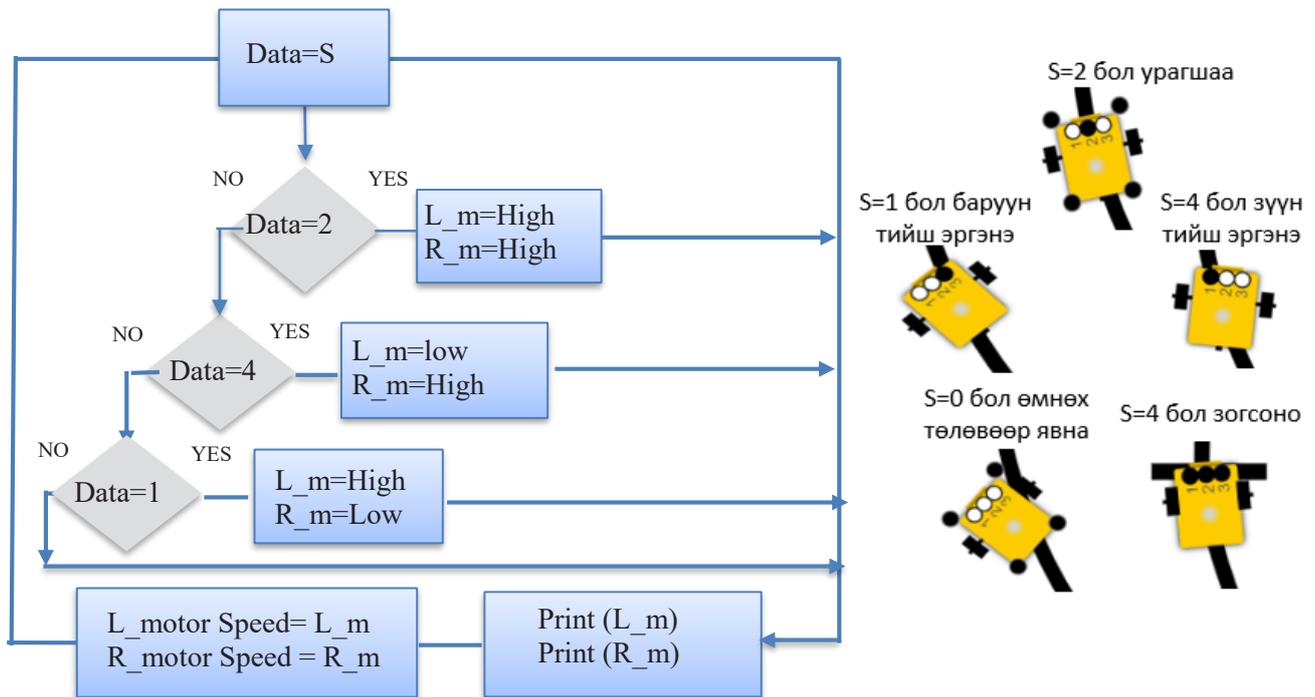
+ start show sensor data
- variable S : Number ← 0
- variable Data_reading : Number ← 0

repeat indefinitely
do
set S to 0
+ if get colour colour sensor Port 1 = 
do
set S to S + 4
+ if get colour colour sensor Port 2 = 
do
set S to S + 2
+ if get colour colour sensor Port 3 = 
do
set S to S + 1
set Data_reading to S

```

| Хоёртын тоолол | Аравтын тоолол | Хийх үйлдэл $2^2=4$ |
|----------------|----------------|---------------------|
| $2^3 2^2 2^1$ | Data=S | Хувьсагч |
| 0 0 0 | 0 | Өмнөх төлөв хадгал |
| 0 0 1 | 1 | БАНУУН тийш эргэ |
| 0 1 0 | 2 | УРАГШАА яв |
| 0 1 1 | 3 | |
| 1 0 0 | 4 | ЗҮҮН тийш эргэ |
| 1 0 1 | 5 | |
| 1 1 0 | 6 | |
| 1 1 1 | 7 | ЗОГС |

Гурван мэдрэгчтэй зураас дагагч роботод шаардлагатай тоон битүүдыг ялган авч тухайн бит болон аравтын тоон утагд харгалзах роботын үйлдлийг тооцоолцгооёе.



Open Roberta Sim E.V3 ийн симуляцийн орчинд гурван мэдрэгчтэй зураас дагач роботын програмыг алгоритмын програмын кодыг бичигчгээе.

```
+ start
+ show sensor data
- variable S : Number ← 7
- variable High_Speed : Number ← 100
- variable LoW_speed : Number ← 50
- variable L_motor : Number ← 0
- variable R_motor : Number ← 0
- variable Data : Number ← 0

repeat indefinitely
do
  set S to 0
  + if [get colour colour sensor Port 1] = [black]
  do
    set S to [S + 4]
  + if [get colour colour sensor Port 2] = [black]
  do
    set S to [S + 2]
  + if [get colour colour sensor Port 3] = [black]
  do
    set S to [S + 1]
  set Data to S
  + - if [Data] = [2]
  do
    set L_motor to High_Speed
    set R_motor to High_Speed
  else if [Data] = [4]
  do
    set L_motor to LoW_speed
    set R_motor to High_Speed
  else if [Data] = [1]
  do
    set L_motor to High_Speed
    set R_motor to LoW_speed
  else
  steer forwards speed % left [L_motor]
  speed % right [R_motor]
```

1.4.3. Технологи, зургийн хичээлээр сурсан мэдлэгээ робот зохион бүтээхэд хэрхэн ашиглах вэ?

Open Roberta Sim E.V3 ын зураас дагагч роботын явах талбарыг зурж, системд оруулахад дараах алхмуудыг хийх хэрэгтэй.

1. Замын төлөвлөгөө гаргах

Алхам 1: Юуны өмнө, роботын зам ямар байхыг төсөөлөөд зураг гаргаарай. Жишээ нь, шулуун шугам, эргэлттэй зам, эсвэл салаа замтай байж болно.

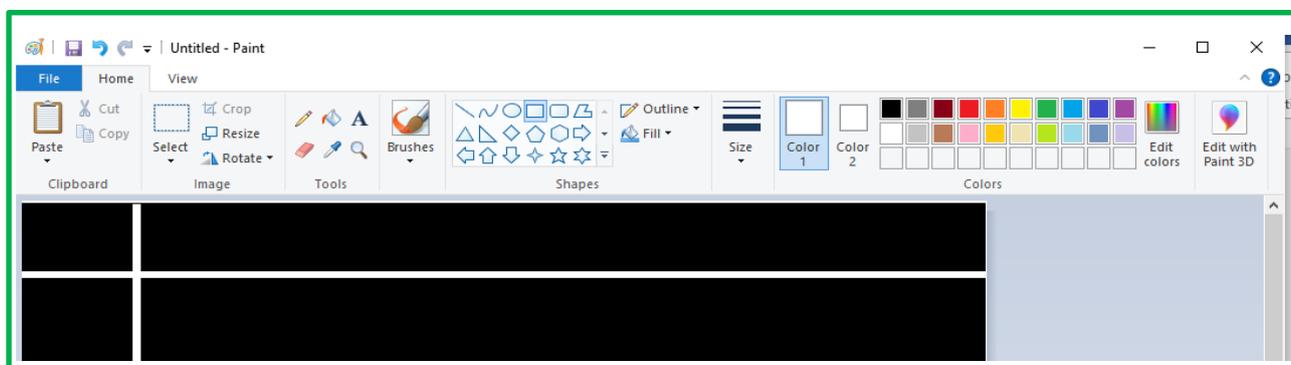
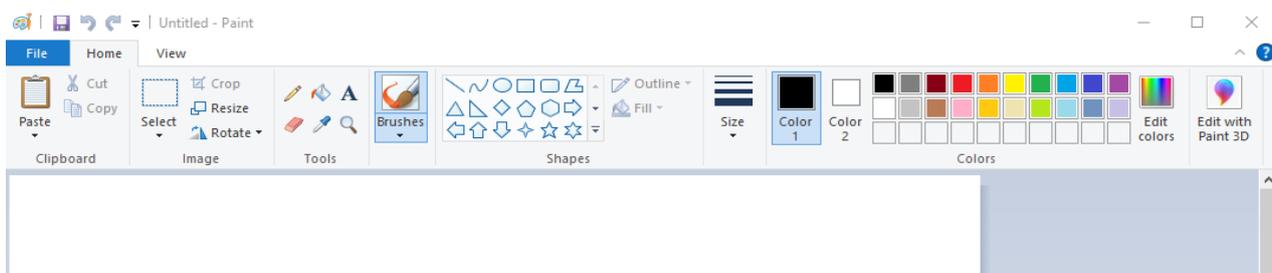
Алхам 2: Талбайн хэмжээ, шугамын өргөн зэргийг төлөвлө. Зураас дагагч роботын хувьд ихэвчлэн 2-3 см өргөнтэй шугам тохиромжтой.

2. Зам зурах програм ашиглах

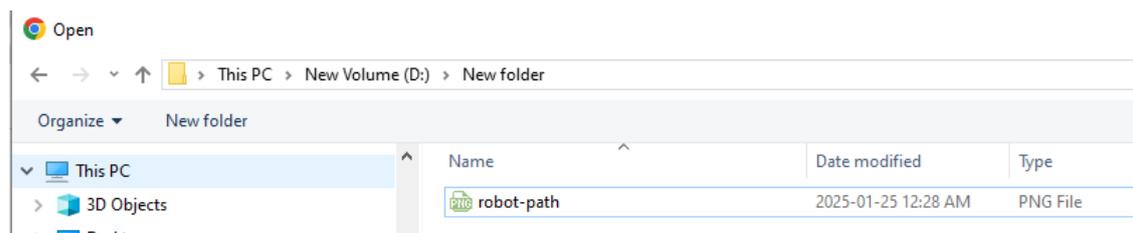
Замын зургийг бүтээхийн тулд Microsoft Paint (эсвэл Paint 3D) ашиглаж болно.

Програмыг нээж, хоосон зургийн хуудсыг сонгоорой.

- ❖ Замын үндсэн хэлбэрийг цагаан өнгийн шугамаар зур.
- ❖ Хар дэвсгэр дээр зам тод харагдах ёстой.
- ❖ Замын эхлэл, төгсгөл болон замын саадыг нэмэхийн тулд дүрс зурах хэрэгслийг ашигла.



- ❖ Зурсан файлаа **JPEG** эсвэл **PNG** хэлбэрээр хадгалах.
- ❖ Хадгалахдаа файлд ойлгомжтой нэр өг (жишээ нь, robot-path.png).

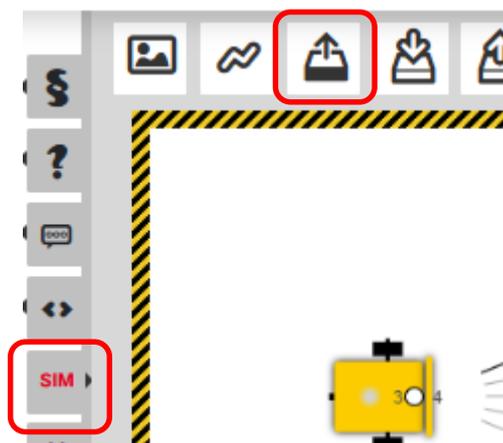
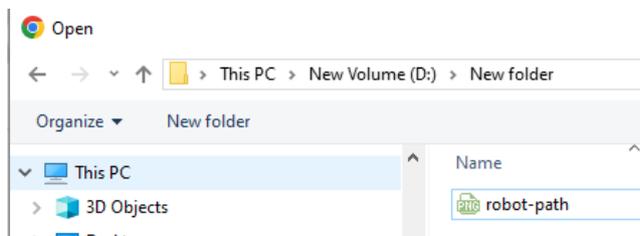


3. Зургаа Open Roberta-д оруулах

- ❖ Вэб хөтөч дээрээ Open Roberta Lab нээх (<https://lab.open-roberta.org>).

Симулятор хэсэг рүү орж "Sim" (Simulator) товчийг дарж, симуляторыг нээ.

- ❖ "Choose File" товч дээр дарж зурсан замын файлаа сонго.



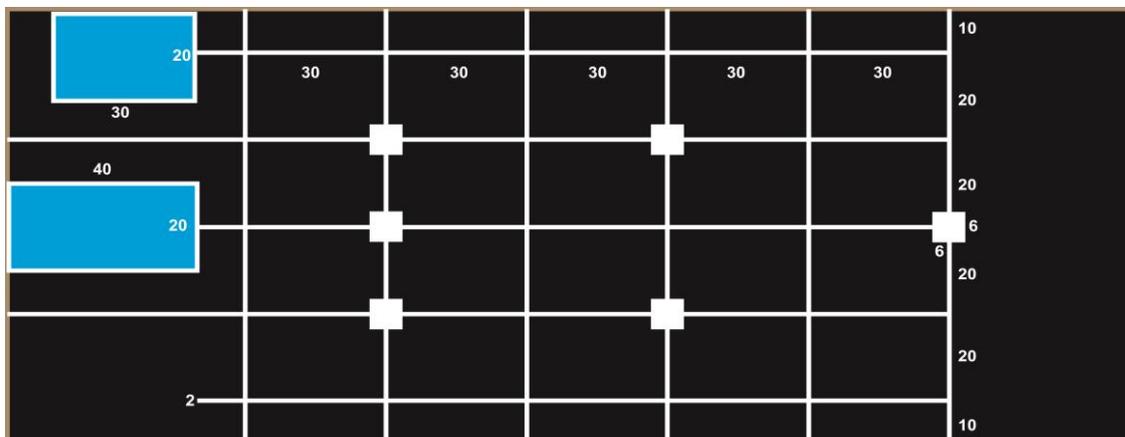
- ❖ Зургаа ачаалсны дараа зам тань симулятор дээр харагдана.



Одоо бүгдээрээ сурсан мэдлэгээ ашиглан “Шоо цуглуулагч робот” тэмцээний талбарыг зурж үзэцгээе.

Тэмцээний талбар:

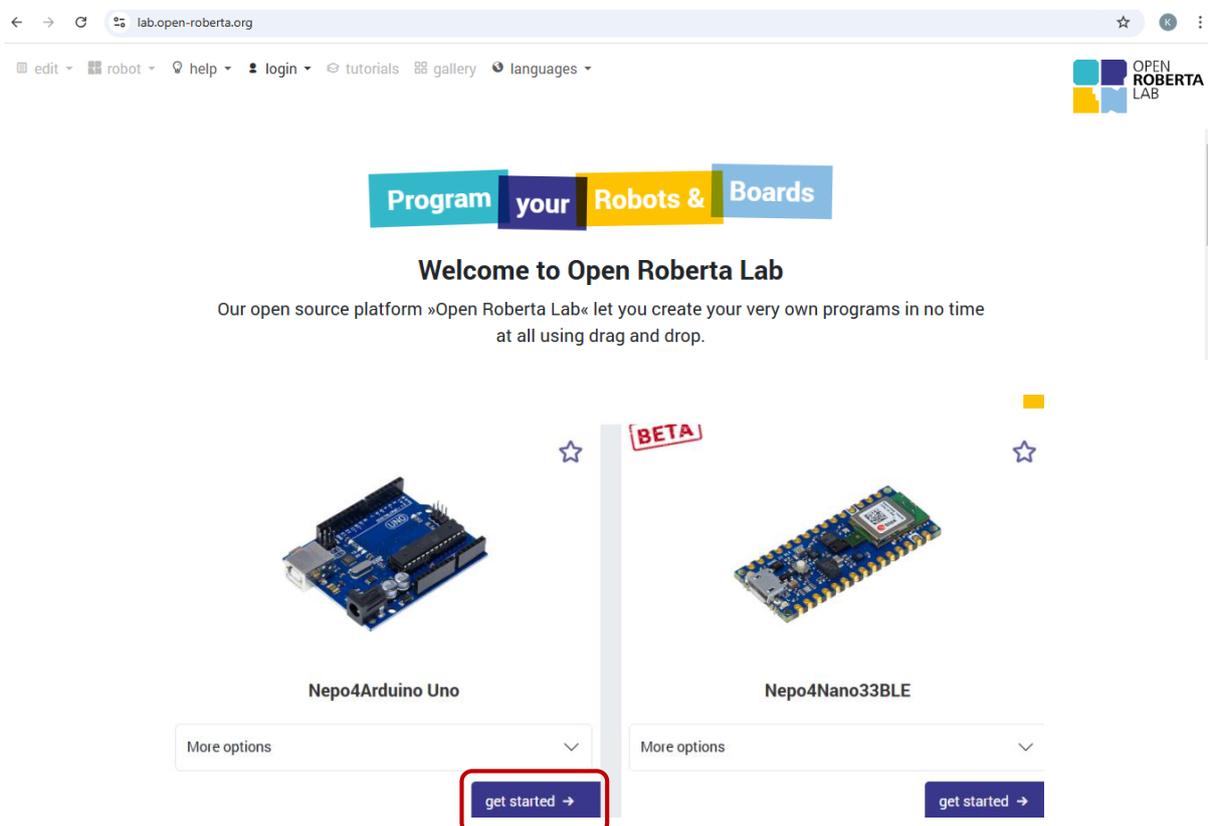
- Зураасны өргөн: 2 см
- Талбайн хэмжээ: 200x240 см (өргөн: урт)
- Роботын эхлэх бүс: 20x30 см(өргөн: урт)
- Шоо цуглуулах бүс: 20x40 см (өргөн: урт)



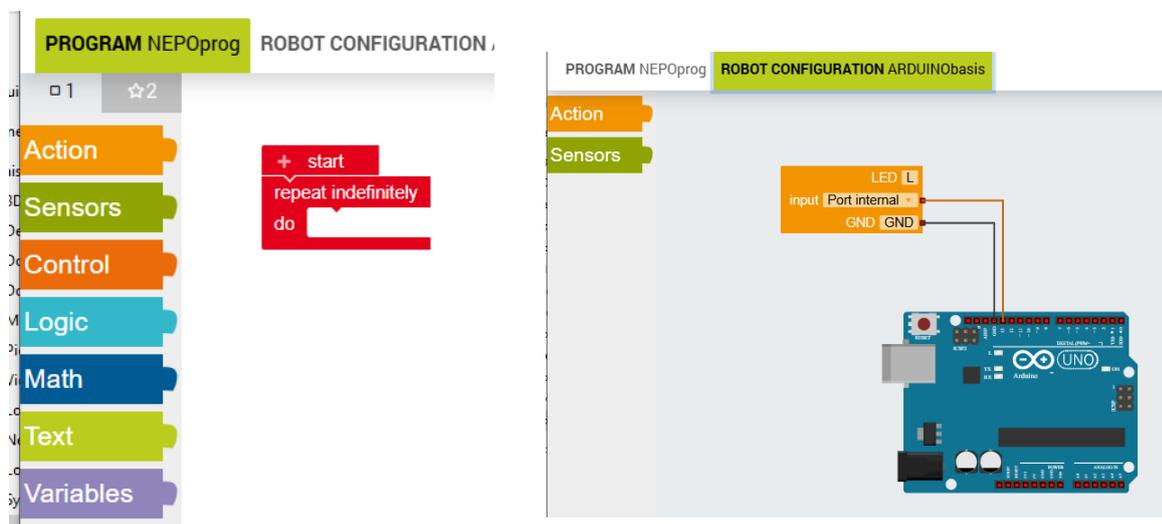
Бүлэг 2. Симуляцийн орчноос Ардуино програмчлалын орчинд



Бид энэ бүлэгт Нээлттэй- Роберта лабораторийн / <https://lab.open-roberta.org/> **Непо4Аrdиuno Uno** симуляцийн талбарыг ашиглан блок програмыг орчинд хялбар бичигдсэн програмын кодыг Ардуино програмчлалын талбарт хөрвүүлэх, тухайн кодыг Ардуино хавтанд шарж бодит төхөөрөмжийг удирдах аргад суралцах болно.



Нээлттэй-Роберта лабораторийн /<https://lab.open-roberta.org/> сайт руу орж **Непо4Аrdиuno Uno** симуляцийн талбарыг сонгоно. Нээлттэй- Роберта лабораторийн **Непо4Аrdиuno Uno** ийн симуляцийн талбар нь *блок програмчлалын PROGRAM NEPOprog* ар болон *оролт, гаралтын төхөөрөмжийн ROBOTCONFIGURATION ARDIUNObasis* талбар гэсэн хоёр хэсгээс бүтнэ.



2.1. Симуляцийн орчинд физик загварыг хэрхэн үүсгэх вэ?

Бид өмнөх бүлэгт Нээлттэй- Роберта лабораторийн орчинг ашиглан оролт , гаралтын төхөөрөмжүүдийг програмчлах, өөрсдийн программыг виртуал робот дээр туршиж үзэх зэргээр физик тоног төхөөрөмж ашиглахгүйгээр хэрхэн туршилт хийх, суралцаж болохыг мэдэж авсан.

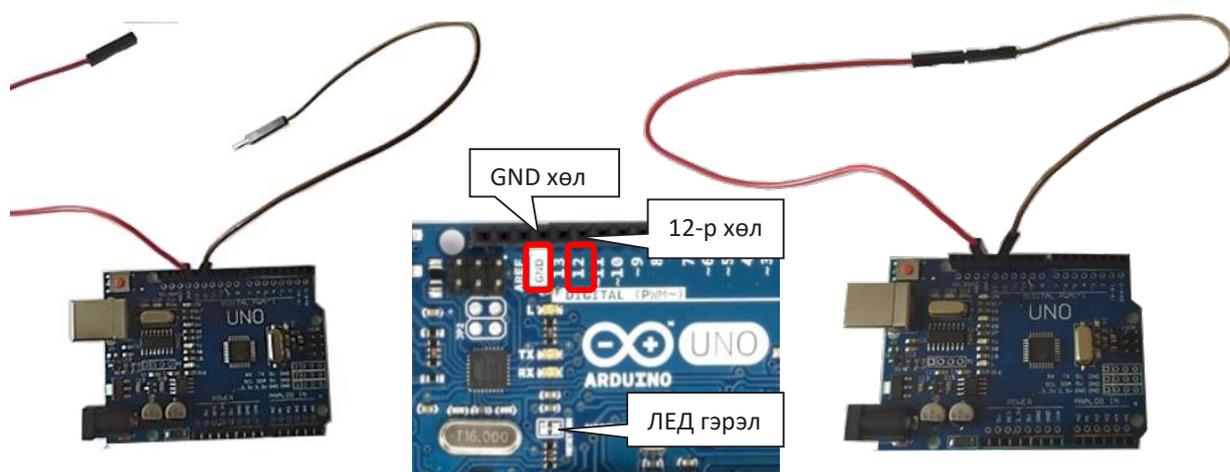
Иймд одоо бид Нээлттэй- Роберта лабораторийн орчны Ардуино блок програмын кодыг Ардуино С програмчлалын орчинд програмын кодыг хөрвүүлж, бодит төхөөрөмжийн програмын үр дүнг турших аргад суралцахад дараах алхмуудыг хийнэ. Үүнд:

- A. Ардуино төхөөрөмжийн гүйцэтгэх үйлдэл, даалгаврыг тодорхойлох
- B. Нээлттэй- Роберта лабораторийн орчны **Nepo4Arduino Uno** симуляцийн талбарт өгөгдсөн даалгаварт тохируулан удирдлагын төхөөрөмжинд холбогч товчлуур болон гэрэлний холболтыг хийж загварчлах
- C. Өгөгдсөн алгоритмын дагуу **Nepo4Arduino Uno** симуляцийн талбарт блок програмын кодыг бичиж, Nepo4Arduino C кодруу хөрвүүлэх
- D. Ардуино хавтанг USB кабель ашиглан хэрэглэгчийн компьютертэй холбох, төхөөрөмжийн холболтыг шалгах
- E. Ardiuno IDE програмын талбарыг нээж, Ардуино хавтангийн холболт болон ажиллагааг шалгах
- F. Нээлттэй- Роберта лабораторийн орчны Nepo4Arduino талбарын C програмын кодыг Ardiuno IDE програмын талбарт кодыг хуулж хөрвүүлэх
- G. Ardiuno IDE програмын талбараас Ардуино хавтанд програмыг шарж ажиллуулах

A. Ардуино төхөөрөмжийн гүйцэтгэх үйлдэл, даалгаврыг тодорхойлох

Нээлттэй- Роберта лабораторийн орчны Nepo4ArduinoUno симуляцийн талбарын програмын кодыг бодит Ардуино Уно хавтангийн програмын код руу хөрвүүлж, төхөөрөмжийг ажиллуулах даалгавруудыг гүйцэтгэцгээе.

Даалгавар 2.1. Ардуино хавтангийн 12 -р портыг GND тэй холбох үед ЛЕД гэрэл унтарч, харин салгах үед гэрэл асах туршилтыг хийцгээе.

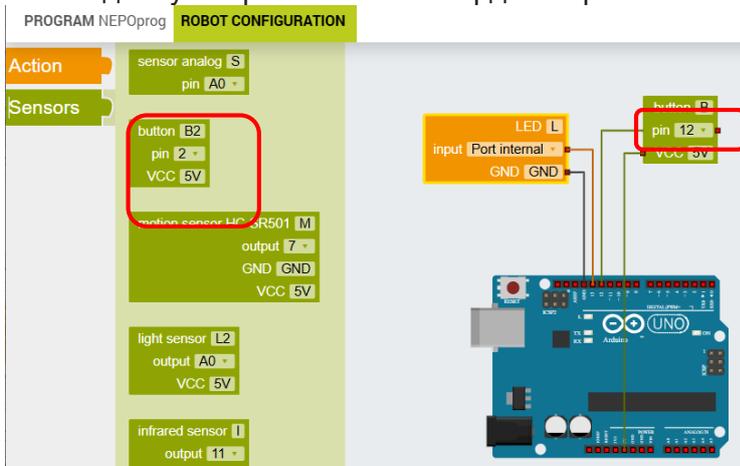


Зураг 2.1 Ардуино хавтангийн GND болон 12-р порт уруу холбосон байдал

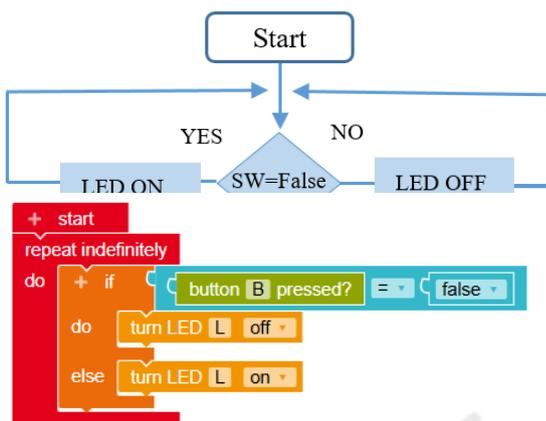
B. Нээлттэй- Роберта лабораторийн орчны Nepo4Arduino Uno симуляцийн талбарт өгөгдсөн даалгаварт тохируулан удирдлагын төхөөрөмжинд холбогч Товчлуур болон гэрлийн холболтыг хийж загварчлах

Нээлттэй- Роберта лабораторийн орчны **Nepo4Arduino Uno** симуляцийн талбарт **Оролт, гаралтын төхөөрөмжийн ROBOTCONFIGURATION ARDIUNObasis** **ROBOT CONFIGURATION ARDUINObasis** хэсэгт дараах холболтуудыг хийнэ.

Энд Товчлуур (Button)  блокыг чирэн холболтын талбарт оруулаад, pin бичвэрийг 12 болгож өөрчилснөөр Ардуино хавтангийн 12-р хөлтэй товчлуур холбогдох болно. Харин LED гэрэл нь талбарын дотоод горимоор холболт хийгдсэн байдаг тул өөрчлөлт хийх шаардлагагүй.



C. Өгөгдсөн алгоритмын дагуу Nero4Arduino Uno симуляцийн талбарт блок програмын кодыг бичиж, Nero4Arduino C кодруу хөрвүүлэх



```

1 // This file is automatically generated by the Open
2
3 #include <Arduino.h>
4
5 #include <NEPODefs.h>
6
7
8 int _taster_B = 12;
9 int _led_L = LED_BUILTIN;
10 void setup()
11 {
12     pinMode(_taster_B, INPUT);
13     pinMode(_led_L, OUTPUT);
14 }
15
16 void loop()
17 {
18     if ( digitalRead(_taster_B) == false ) {
19         digitalWrite(_led_L, LOW);
20     } else {
21         digitalWrite(_led_L, HIGH);
22     }
23 }

```

D. Ардуино хавтанг USB кабель ашиглан хэрэглэгчийн компьютертэй холбох, төхөөрөмжийн холболтыг шалгаж, ажиллуулах

- Ардуино хөгжүүлэлтийн програмыг ажиллуулахын өмнө USB кабель ашиглан Ардуино хавтанг хэрэглэгчийн компьютертэй холбох



Зураг 2.2 Ардуиног компьютертэй холбосон байдал

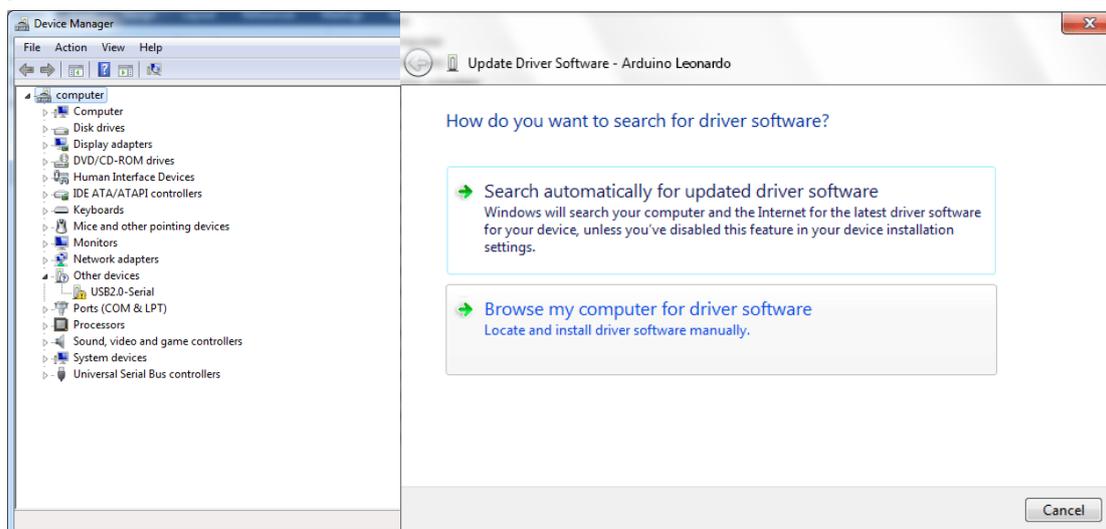
- USB кабелийн холболтын мэдээллийг дэлгэцэн дээр гарч иртэл түр хүлээх
- Хэрэв USB холболтын төхөөрөмжийг компьютер танихгүй бол түүний драйвер програмыг сонгон уншуулж, таниулах



USB холболтын төхөөрөмжийг таниулах Control Panel\All Control Panel Items\System

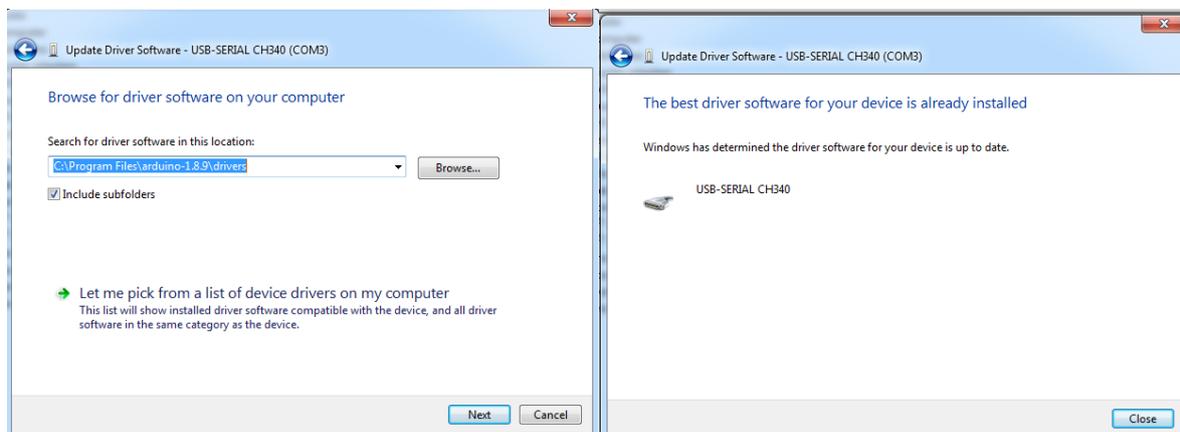
Хэрэв USB кабелийн драйверыг уншуулж, таниулах процесс автоматаар хийгдэхгүй тохиолдолд дараах алхмуудыг хийнэ.

- Control Panel\All Control Panel Items\System замаас Device Manager (Start> Control Panel>System) хавтсыг сонгон хулганы зүүн товчлуурыг дарна. Device Manager хавтасны Other device цэснээс Update Driver Software-ыг сонгоно.
- Дараа нь “Browse my computer for driver software” цэсийг сонгоно.



Зураг 2.3 USB кабелийг автоматаар таниагүй тохиолдолд драйверын замыг зааж, сонголтийг хийх

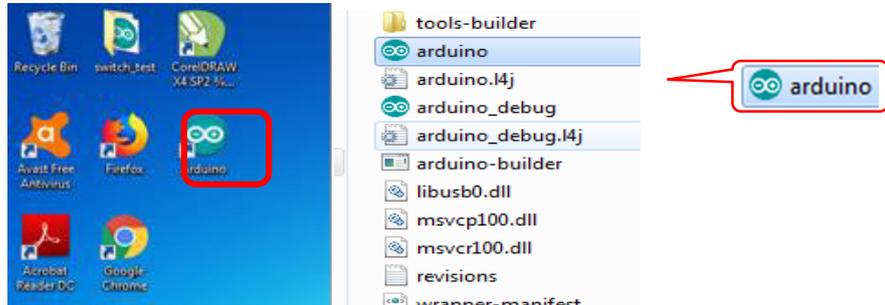
- “Browse ...” товчийг идэвхжүүлэх ба дэлгэцэнд харагдах цэснээс Ардуино хавтантай холбосон USB кабелийн драйвер нь байрлах замыг зөв сонгон “NEXT” товч дарна.
- Ийнхүү хэрэглэгчийн компьютер болон Ардуино төхөөрөмж алдаагүй холбогдсоныг илэрхийлэх бичвэр гарснаар Ардуино хөгжүүлэлтийн програмыг ашиглах бэлтгэл ажил хангагдана.



Зураг 2.4 Компьютерын систем нь USB кабелийг бүрэн танысан байдал

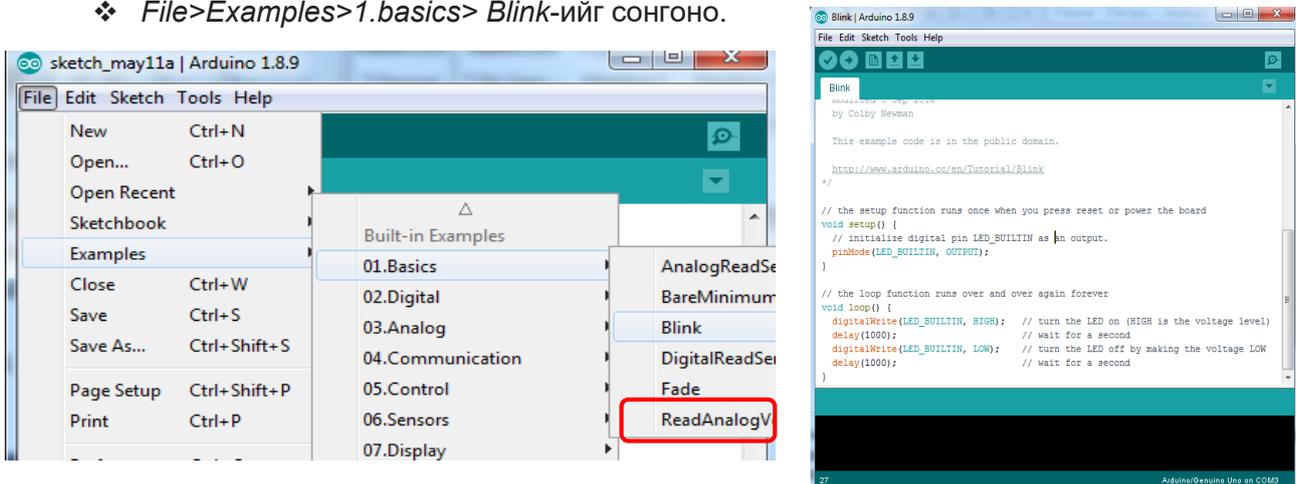
Е. Arduino IDE програмын талбарыг нээж, Ардуино хавтангийн холболт болон ажиллагааг шалгах

Desktop дээрх  shortcut эсвэл C:\Program Files\arduino-.... хавтаснаас програмыг дуудаж ажиллуулна.

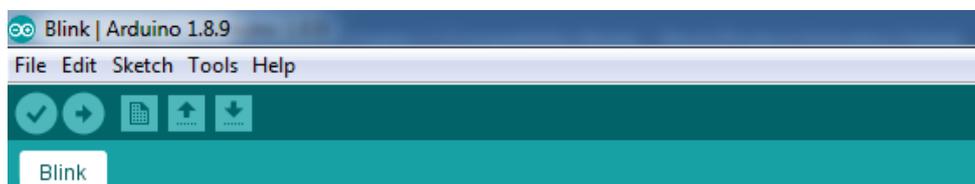


Ардуино хавтанд ямар нэгэн нэмэлт төхөөрөмж холбохгүйгээр төхөөрөмжийн хэвийн ажиллагааг хянах зориулалттай ЛЕД (*LED*) гэрлэн диод байдаг. ЛЕД гэрлийг ашиглан Ардуино хөгжүүлэлтийн санд байгаа Blink нэртэй програмын тусламжтайгаар төхөөрөмжийн ажиллагааг хянах боломжтой. Үүнийг дараах дарааллаар гүйцэтгэнэ.

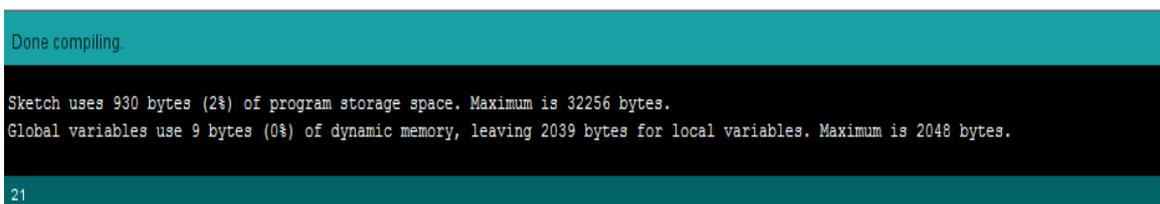
❖ *File>Examples>1.basics> Blink*-ийг сонгоно.



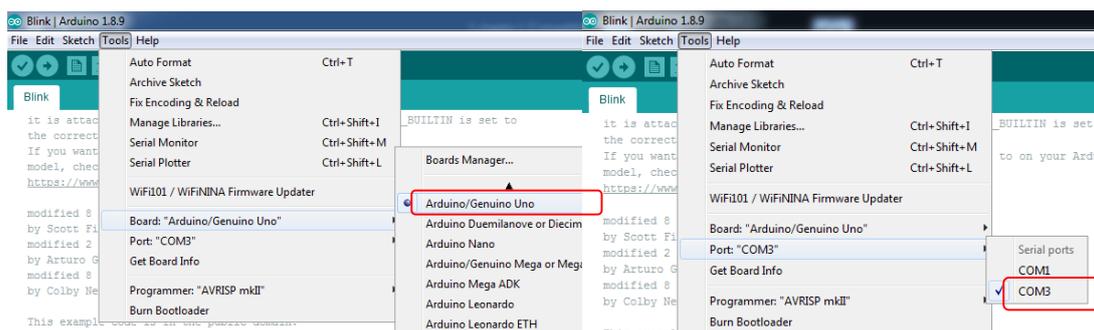
Ардуино хөгжүүлэлтийн програмын зүүн дээд буланд  байгаа товчлуурыг дарж програмын кодыг хөрвүүлэх үйлдлийг хийнэ.



Ардуино хөгжүүлэлтийн програмын доод хэсэгт нь “Done compiling” гэсэн бичигтэй зационд хөрвүүлэлт амжилттай болсныг илэрхийлсэн дараах текст гарч ирнэ.



Зураг 2.5-д Ардуино УНО хавтангийн төрлийг програм дээрээс сонгон, компьютертэй цуваа (*serial*) холболт хийх тохиргоог харуулав.

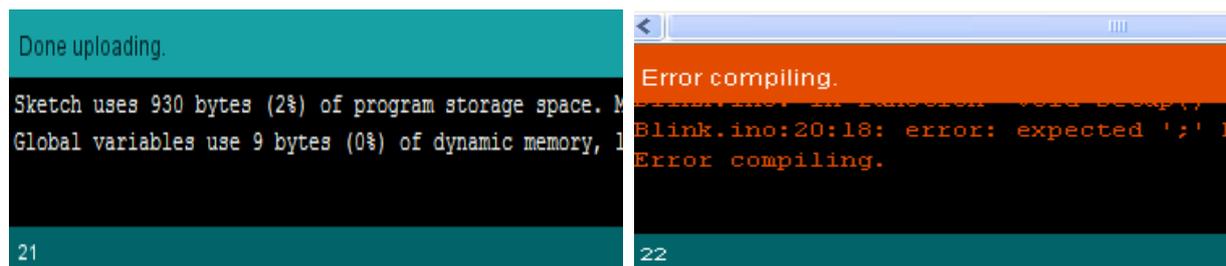


Зураг 2.5 Ардуино хөгжүүлэлтийн орчны програмын тохиргоо

Ардуино програмын файлыг хөгжүүлэлтийн хавтанд суулгахыг ачаалах гэнэ. Ачаалалтыг баруун тийш заасан  сум бүхий товчлуураар эхлүүлнэ.

Програм болон төхөөрөмжийн ажиллагаа, холболтууд алдаагүй тохиолдолд ажиллагааны процесс, алдааны мэдээллийг харуулах талбар дээр *Done compiling* гэсэн бичиг гарснаар Ардуино хавтан дээрх ЛЕД гэрэл анивчих болно.

Харин ямар нэгэн алдаа гарсан тохиолдолд *Error compiling*г эсэн бичиг гарч төхөөрөмж ажиллахгүй.



Зураг 2.6 Хавтангийн ажиллагааг тодорхойлох цонхнууд

Зааврын дагуу суулгалт амжилттай болсон тохиолдолд хавтан дээр байрлах ЛЕД гэрэл нь нэг секундйн давтамжтайгаар анивчих болно.



Зураг 2.7 Хавтангийн гэрэл анивчиж буй байдал

Ф. Нээлттэй- Роберта лабораторийн орчны Nero4Arduino талбарын C програмын кодыг Arduino IDE програмын талбарт кодыг хуулж хөрвүүлэх Arduino IDE програмын талбар дахь кодны хуулж програмын кодноос `#include <NEPODefs.h>` мөрийг устгаарай.

```

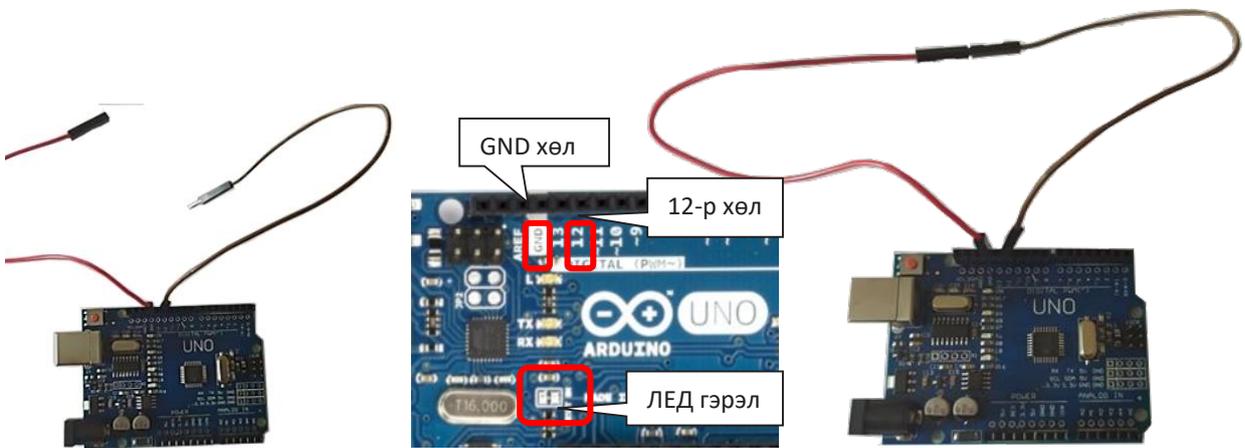
1 // This file is automatically generated by Arduino IDE 2.3.2
2
3 #include <Arduino.h>
4
5 #include <NEPODefs.h>
6
7
8 int _taster_B = 12;
9 int _led_L = LED_BUILTIN;
10 void setup()
11 {
12     pinMode(_taster_B, INPUT);
13     pinMode(_led_L, OUTPUT);
14 }
15
16 void loop()
17 {
18     if ( digitalRead(_taster_B) == false )
19         digitalWrite(_led_L, LOW);
20     } else {
21         digitalWrite(_led_L, HIGH);
22     }
23 }

```

```

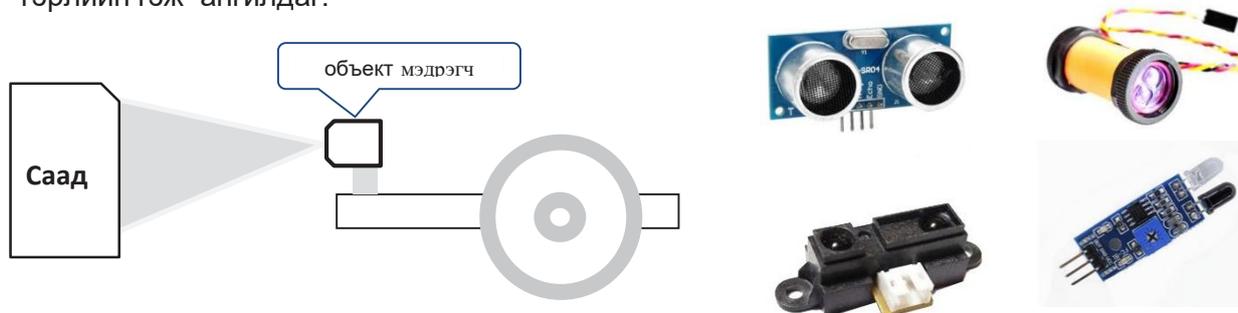
1 // This file is automatically generated by Arduino IDE 2.3.2
2
3 #include <Arduino.h>
4
5 int _taster_B = 12;
6 int _led_L = LED_BUILTIN;
7 void setup()
8 {
9     pinMode(_taster_B, INPUT);
10    pinMode(_led_L, OUTPUT);
11 }
12
13 void loop()
14 {
15     if ( digitalRead(_taster_B) == false )
16         digitalWrite(_led_L, LOW);
17     } else {
18         digitalWrite(_led_L, HIGH);
19     }
20 }

```

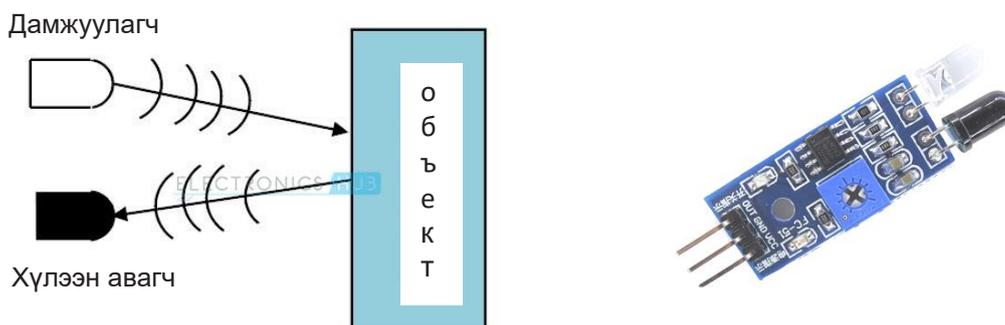


2.2 Мэдрэгч, түүний програмчлал

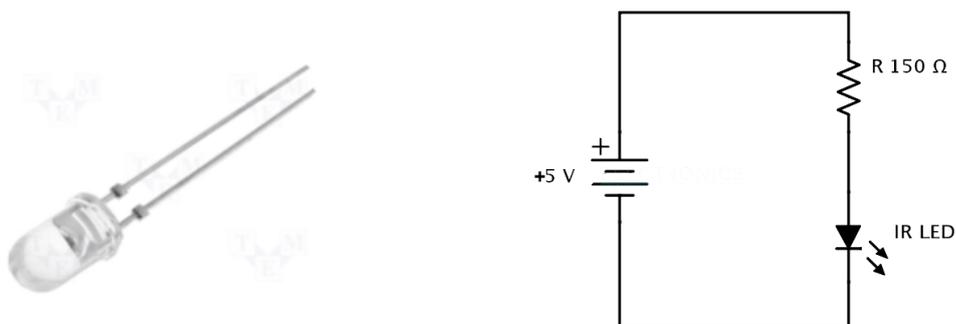
Объект мэдрэгч нь тухайн биетээс заагдсан объект хүртэлх зайг дохио дамжуулагч болон хүлээн авагч төхөөрөмжийн тусламжтайгаар тодорхойлдог бөгөөд хийц хэлбэр, технологиосоо хамааран олон төрөл байдаг. Мөн төхөөрөмжийн дамжуулагч дохионы шинж чанараас хамааран имфра улаан туяаны, радио долгионы, гэрлийн гэж ангилдаг.



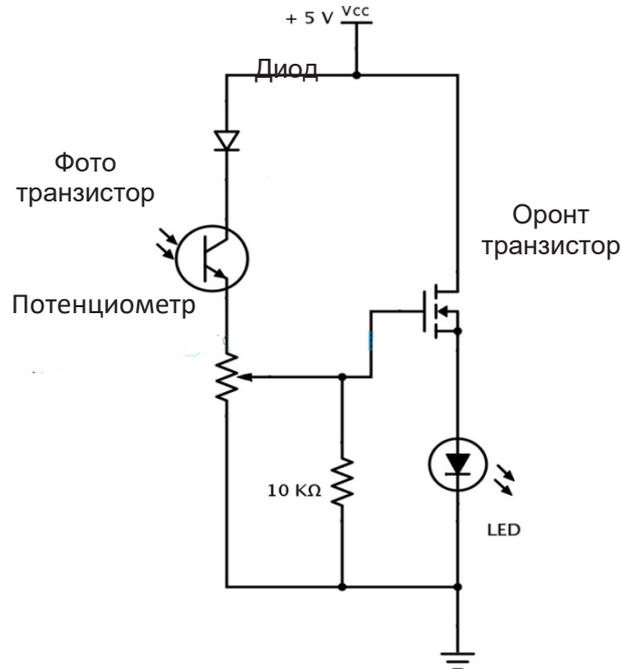
Энд объект мэдрэгч төхөөрөмжийн хамгийн энгийн шийдэл болох имфра улаан туяаны (IR) мэдрэгчийг танилцуулж байна. Энэ төхөөрөмж нь гэрлийн долгион дамжуулагч болон хүлээн авагч гэсэн хоёр хэсгээс бүрддэг бөгөөд гэрлийн долгион дамжуулагчаас цацагдсан дохио нь объектод ойж хүлээн авагчид ирэх үеийн дохионы түвшинг ашиглан тухайн биетээс объект хүртэлх зайг тодорхойлдог.



Имфра улаан гэрлийн дамжуулагч нь имфра улаан туяаг цацруулдаг гэрлэн диод байдаг бөгөөд үүнийг IR LED гэж нэрлэдэг. IR LED нь энгийн гэрлэн диодтой ижил ажиллагаатай боловч хүний нүдэнд үл үзэгдэх туяаг цацруулдагаараа онцлог юм. IR LED болон түүний хэлхээн дэх холболтыг доорх зурагт харууллаа. IR дамжуулагчийн хэлхээ нь 5 вольтоор тэжээгдэх үедээ ойролцоогоор 3-5 mA гүйдэл хэрэглэдэг байна.



Харин имфра улаан туяаны хүлээн авагчууд нь IR дамжуулагчаас гарсан цацрагийг илрүүлдэг бөгөөд түүнийг имфра улаан туяаны мэдрэгч буюу IR хүлээн авагч гэж нэрлэдэг. IR хүлээн авагчууд нь ихэвчлэн фотодиод эсвэл транзистор байдаг. Хэлхээний хүчдэл, хийц хэлбэр, долгионы урт гэх мэт зүйлүүдээсээ хамааран IR хүлээн авагчууд нь олон төрөл байдаг бөгөөд имфра хүлээн авагч болон дамжуулагчийг холбон хэрэглэх үед хүлээн авагчийн долгионы урт нь дамжуулагчийнхтай зохицож байх ёстой.



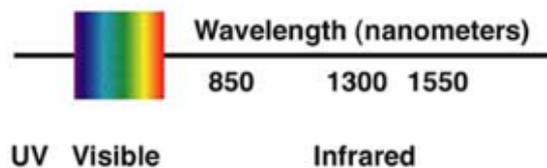
LED, потенциометр, оронт транзистор (MOSFET), диод, фототранзисторээс бүтсэн IR хүлээн авагчийн хэлхээний схемийг дээрх зурагт харууллаа. Фото транзистор нь ямар нэгэн имфра цацрагийг хүлээн авах үед хэлхээгээр гүйдэл гүйж оронт транзистор нээгдэн ажиллах бөгөөд улмаар ачаан дээрх LED гэрэл асна.

- Цацруулагч: SFH4550 (850 nm)

https://www.rt-shop.jp/index.php?main_page=product_info&products_id=1253



Хүний сонсож чадах дохионы спектр 20Гц-20кгц
Хүний харж чадах долгионы урт 390нм -700 нм
IR мэдрэгчид хэрэглэдэг долгионы урт 850нм- 1550 нм

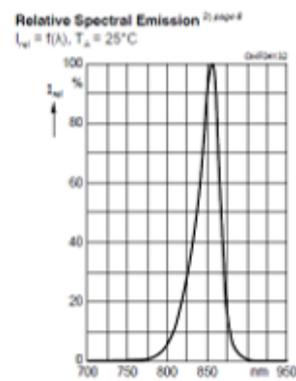


Шинж чанар

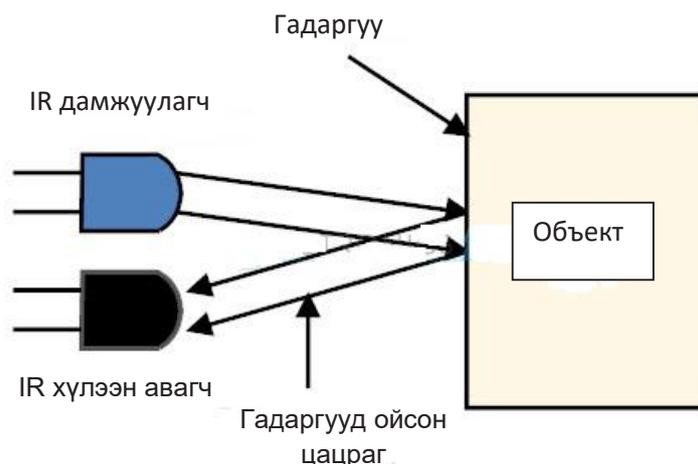
- Чадлын IR LED
- Цацруулах өнцөг $\pm 3^\circ$
- Туяаны цацрагын чадал өндөр

Хэрэглээ

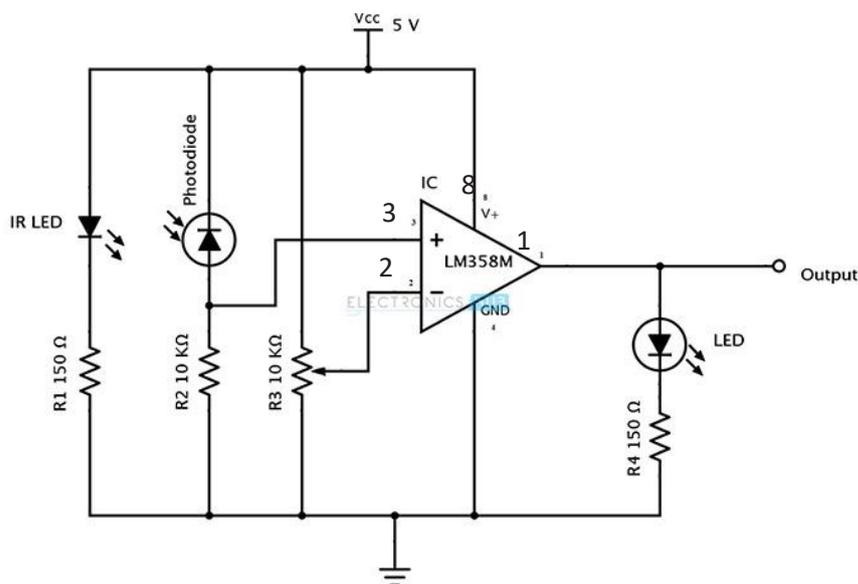
- Шөнийн камер
- Өгөгдөл дамжуулах
- Мэдрэгчийн технологи
- Утааны мэдрэгч



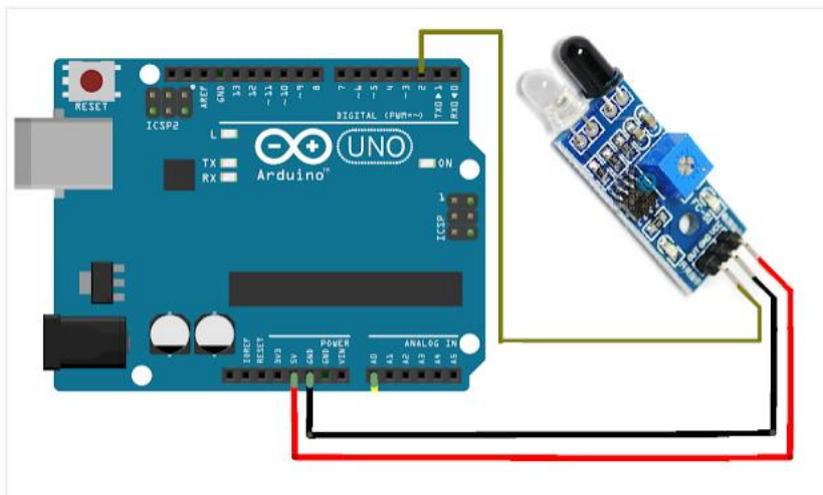
Ифра имфра улаан туяаны дамжуулагч болон хүлээн авагч ашигласан объект мэдрэгчийн ажиллагааны зарчмыг доорх зураг харууллаа.



Объект мэдрэгч нь IR LED болон IR фотодиодоос бүтсэн бөгөөд тэдгээрийн хослолыг Опто-Холбогч гэж нэрлэдэг. IR дамжуулагч нь имфра улаан туяаг цацруулах бөгөөд цацрагийн хэсэг объектод хүрэх үед цацрагын тусгал нь буцаж IR хүлээн авагч руу ойдог. IR хүлээн авагчийн ажиллагаагаа нь имфра улаан туяаны идэвхижилтээс хамааран мэдрэгчийн гаралтыг тодорхойлдог. IR LED, фотодиод, потенциометр, үйлдлийн өсгүүр болон LED гэрлээс бүрдсэн объект мэдрэгчийн схемийг доорх зурагт харуулсан байна. IR LED гэрлээс цацрах имфра улаан туяаны цацраг нь фотодиодон дээр туссанаар фотодиодоны эсэргүүцэл багасаж, улмаар түүнтэй цуваагаар холбогдсон эсэргүүцэл дээрх хүчдэл ихэснэ. Хэлхээнээс харахад үйлдлийн өсгүүрийн нэг оролт нь фотодиод болон эсэргүүлэлтэй холбогдсон бөгөөд нөгөө оролт нь потенциометртэй холбогдсон байна. Иймээс үйлдлийн өсгүүрийн оролтон дээрх эсэргүүцлийн хүчдэл болон потенциометрээр тохируулагдах тулгуур хүчдэл хоёр харьцуулагдаж, үйлдлийн өсгүүрийн гаралтын утгыг тодорхойлно. Тухайн орчны нөхцлөөс хамааруулан потенциометрийн утгыг өөрчлөн тулгуур хүчдэлийг тохируулж болно. Хэрвээ фотодиодтой цуваа холбогдсон эсэргүүцэл дээр унах хүчдэл нь тулгуур хүчдэлийн утганаас их бол үйлдлийн өсгүүрийн гаралт өндөр болж, улмаар үйлдлийн өсгүүрийн гаралтанд холбогдсон LED гэрлэн диод асна. Мэдрэгчийн ажиллагаанд IR LED болон IR хүлээн авагчийн байрлал нь маш чухал байдаг.



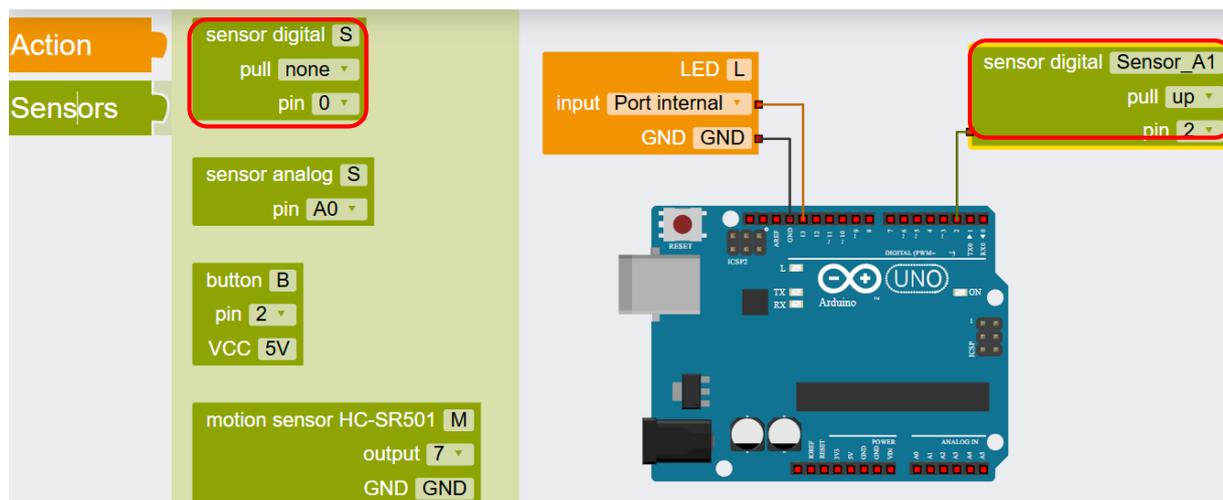
Даалгавар 2.2. Ардуино хавтангийн 2-р хөлөнд IR мэдрэгчийг холбож, мэдрэгчид объект илрэх үед ЛЕД гэрэл асч, харин илрээгүй үед уртрах туршилтыг хийцгээе.



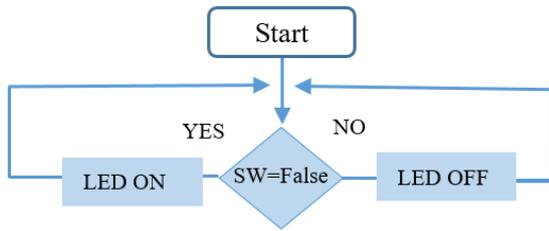
А. Нээлттэй- Роберта лабораторийн орчны Nерo4Arдиuno Uno симуляцийн талбарт өгөгдсөн даалгаварт тохируулан удирдлагын төхөөрөмжинд холбогч Мэдрэгч болон гэрлийн холболтыг хийж загварчлах

Нээлттэй- Роберта лабораторийн орчны **Nерo4Arдиuno Uno** симуляцийн талбарт *Оролт, гаралтын төхөөрөмжийн* ROBOTCONFIGURATION ARDIUNObasis хэсэгт дараах холболтуудыг хийнэ.

Энд Тоон мэдрэгч (Sensor digital) блокыг чирэн холболтын талбарт оруулаад, pull бичвэрийг up, харин pin бичвэрийг 2 болгож өөрчилснөөр Ардуино хавтангийн 2-р хөлтэй мэдрэгч холбогдох болно. Харин LED гэрэл нь талбарын дотоод горимоор холболт хийгдсэн байдаг тул өөрчлөлт хийх шаардлагагүй.



A. Өгөгдсөн алгоритмын дагуу Nерo4Arдиuno Уно симуляцийн талбарт блок програмын кодыг бичиж, Nерo4Arдиuno C кодруу хөрвүүлэх



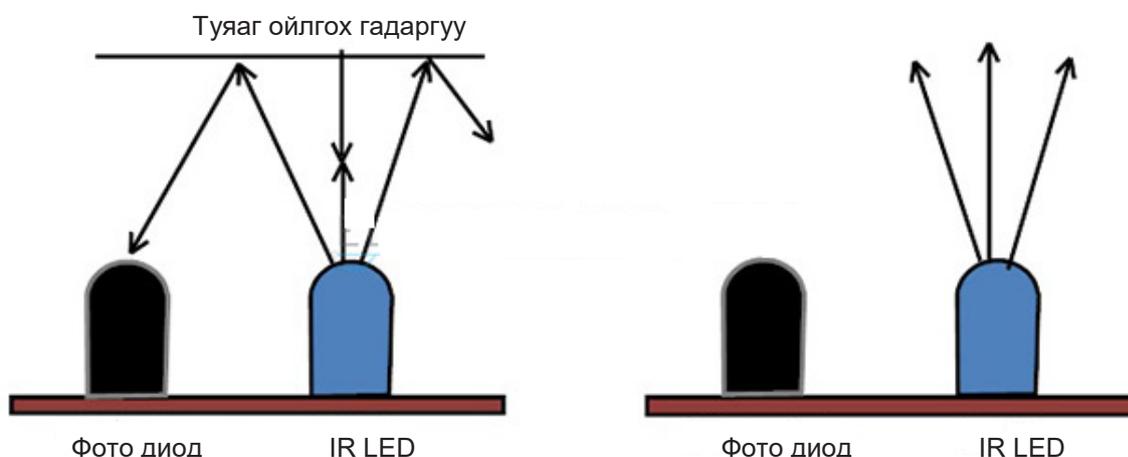
```

1 // This file is automatically generated by the Op
2
3 #include <Arduino.h>
4
5 #include <NEPODefs.h>
6
7
8 int _taster_B = 12;
9 int _led_L = LED_BUILTIN;
10 void setup()
  
```

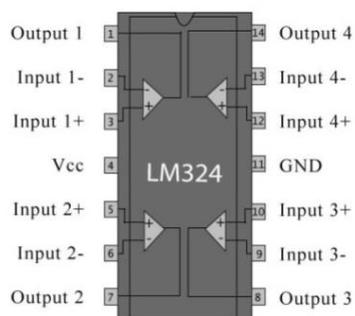
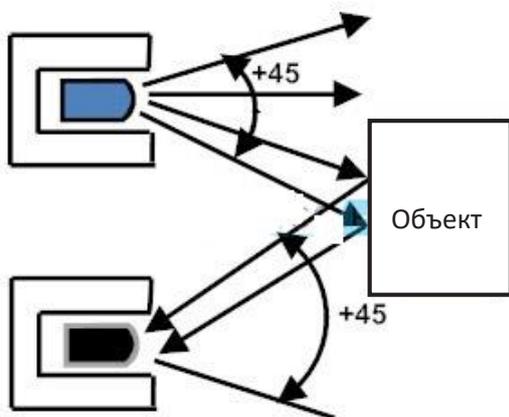
Даалгавар 2.3 . Ардиuno хавтангийн A0 , A1, A2 хөлнүүдэд 3 ширхэг IR мэдрэгчүүдийг холбож, мэдрэгчид объект илрэх үед ЛЕД гэрэл асч, харин илрээгүй үед уртрах туршилтыг хийцгээе.

2.3 Хар цагаан өнгө мэдрэгч, түүний програмчлал

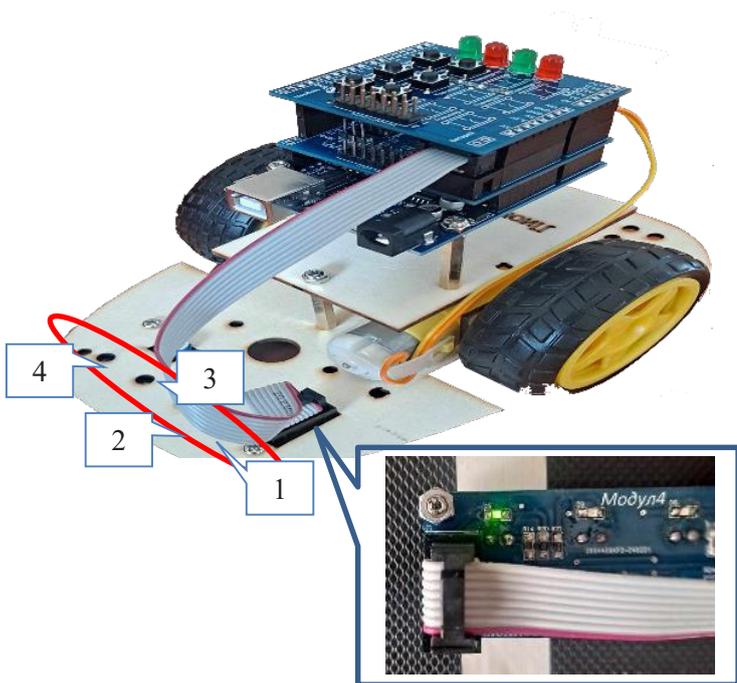
Хар өнгө нь гэрлийн цацрагийг шингээж уусгадаг бол цагаан өнгө нь гэрлийн цацрагийг хугалж ойлгодог бөгөөд физикийн энэ хууль дээр үндэслэн IR LED ба фотодиодийг ашиглан өнгө мэдрэгчийг хийж болдог. Мэдрэгчийн бүтэц нь IR LED ба фотодиод хоёрыг зэрэгцээ байрлуулсан бөгөөд ажиллагаа нь IR дамжуулагчаас түгээгдэх инфра цацрагийн туяа объектыг мөргөж фотодиод руу буцаж ойлх зарчмыг ашигласан. Харин объектын гадаргууг дотор нь туяаг ойлгох болон шингээх гадаргуу гэж хоёр ангилдаг. Хэрвээ объектын гадаргуу нь байгалийн цагаан болон бусад цайвар өнгө байвал түүнийг ойлгох гадаргуу гэх ба цацрагийн эрчимжилтээс хамааран тус объектод ойлсон гэрлийн цацраг фотодиод дээр туссанаар мэдрэгчийн хэлхээгээр гүйдэл гүйдэг. Хэрвээ объектын гадаргуу нь байгалийн хар болон бусад бараан өнгө байвал түүнийг шингээх гадаргуу гэх ба гадаргууд туссан гэрлийн цацраг объектод шингэснээр фотодиодон дээр гэрлийн цацраг тусахгүй тул фотодиодны эсэргүүцэл өндөр хэвээр хадгалах тул хэлхээгээр гүйдэл гүйхгүй. Энэ нь тухайн орчинд ямар нэгэн объект байхгүй байгаатай ижил юм. Зурган дүрслэлийг доор харууллаа.



Мэдрэгчийн ажиллагаанд хүлээн авагч, дамжуулагчийн байрлал маш чухал байдаг. Хүлээн авагч болон дамжуулагч нь хоёулаа тодорхой өнцгөөр байрласан байх ёстой бөгөөд өнцгийг зөв тохируулсан үед объектын илрүүлэлт нь алдаагүй хийгддэг. Дохио хүлээн авагч болон дамжуулагчийг байрлуулахдаа цацрагийн туяаг ± 45 градусын өнцгийн чиглэлтэйгээр дамжуулах болон хүлээн боломжтойгоор тохируулан байрлуулах хэрэгтэй. Чиглэлийг доорх зурагт үзүүлээ. Мөн бид 4 ширхэг өнгийн мэдрэгчийг хийх үед LM324 серийн үйлдлийн өсгүүрийг ашиглах нь илүү тохиромжтой байдаг.

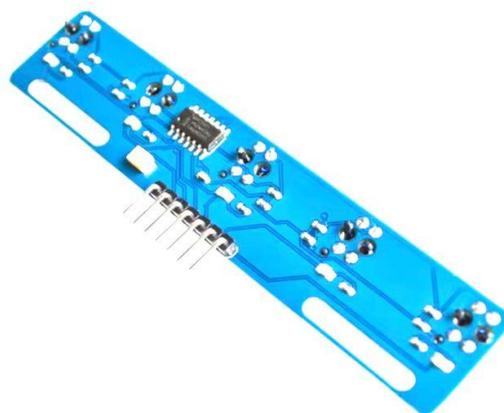
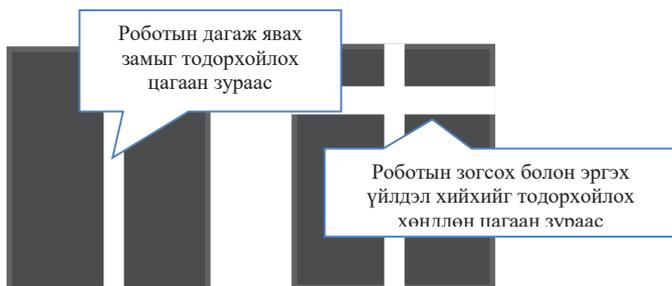


Даалгавар 2.4 . Ардиуно хавтангийн A0 , A1, A2, A4 хөлнүүдэд IR мэдрэгчүүдийг 4 ширхэг тоон гаралтыг холбож, мэдрэгчийн утгыг Аравтын тооллын утга болгох прогармын кодыг бичээрэй.



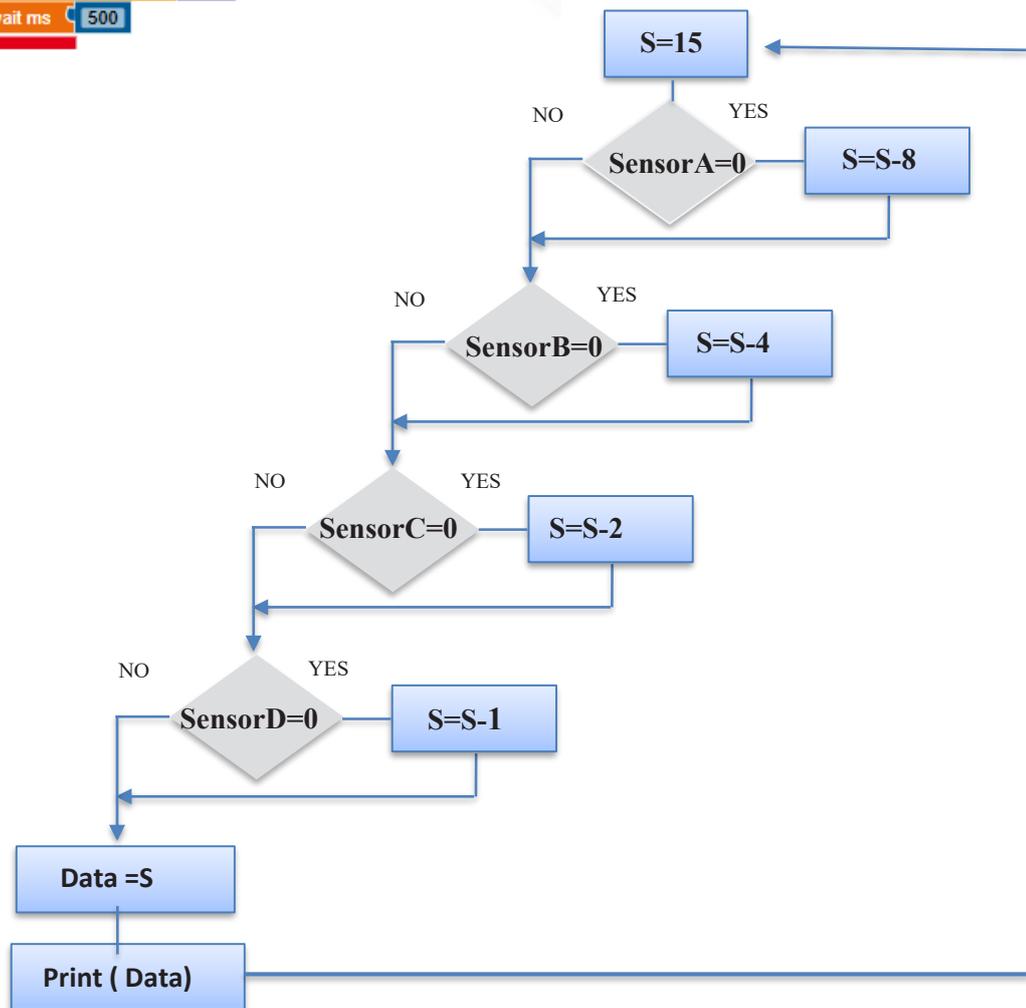
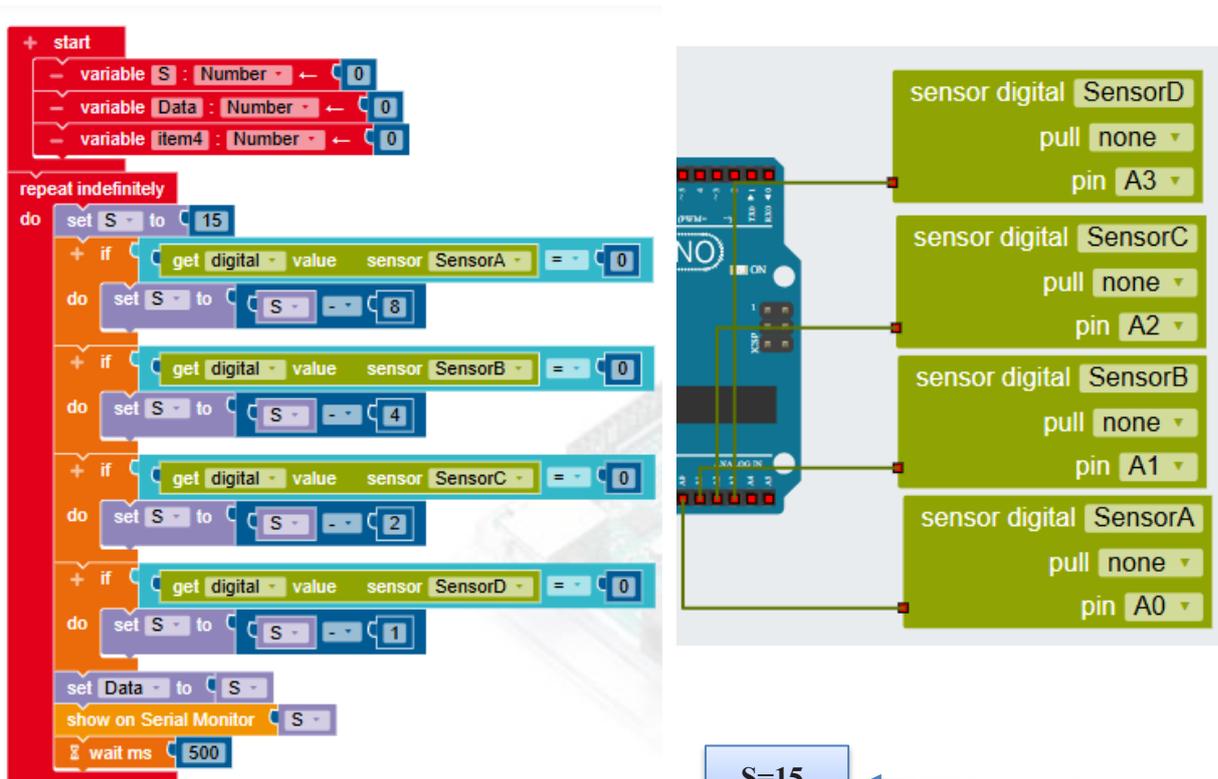
| Аравтын тоолол | Хоёртын тоолол |
|----------------|----------------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| 10 | 1010 |
| 11 | 1011 |
| 12 | 1100 |
| 13 | 1101 |
| 14 | 1110 |
| 15 | 1111 |

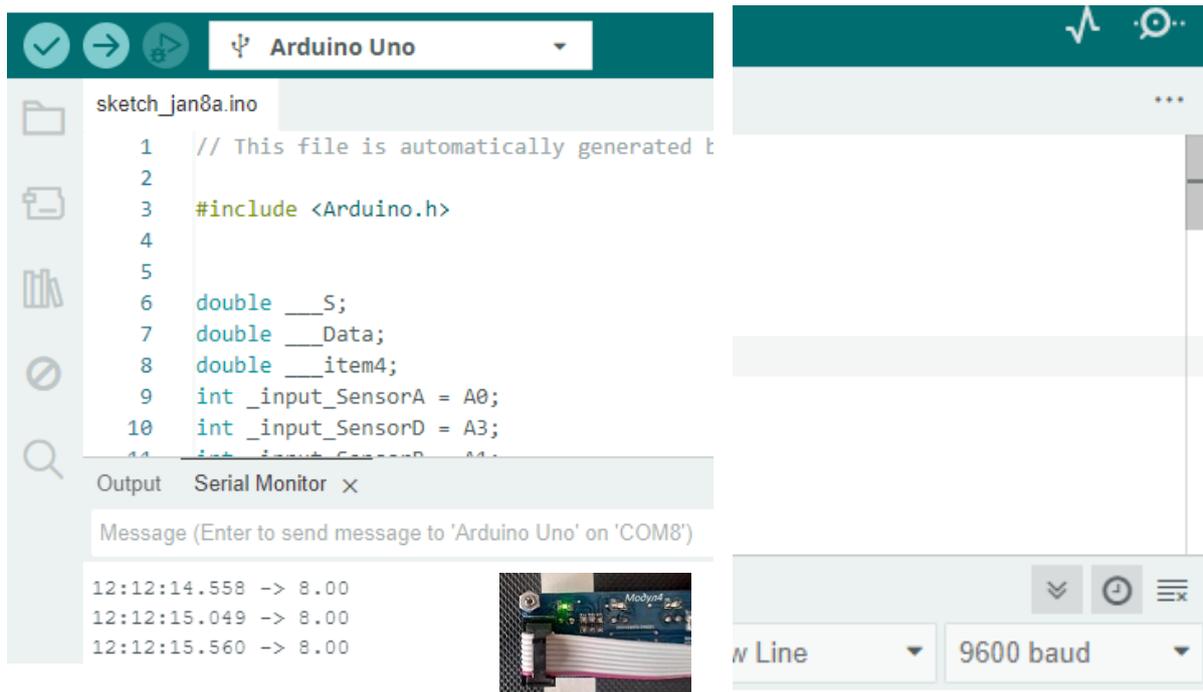
| Интерфейс | Оролт, гаралтын дугаар | Оролт, гаралтын утга | Үйлдэл |
|------------|------------------------|----------------------|----------------------|
| Мэдрэгч A1 | A0 | Оролт =1 бол | Цагаан өнгө мэдэрсэн |
| | | Оролт =0 бол | Хар өнгө мэдэрсэн |
| Мэдрэгч B2 | A1 | Оролт =1 бол | Цагаан өнгө мэдэрсэн |
| | | Оролт =0 бол | Хар өнгө мэдэрсэн |
| Мэдрэгч C3 | A2 | Оролт =1 бол | Цагаан өнгө мэдэрсэн |
| | | Оролт =0 бол | Хар өнгө мэдэрсэн |
| Мэдрэгч D4 | A3 | Оролт =1 бол | Цагаан өнгө мэдэрсэн |
| | | Оролт =0 бол | Хар өнгө мэдэрсэн |



- **IR мэдрэгчүүдийг Ардиуно хавтанд хавтанд холбох дараалал:**
 - IR мэдрэгчийн **OUT** хөлүүдийг Arduino хавтангийн A0, A1, A2, A4 хөлүүдэд холбох.
 - IR мэдрэгчийн **VCC** хөлийг Arduino хавтангийн **5V** холболтод холбох.
 - IR мэдрэгчийн **GND** хөлийг Arduino хавтангийн **GND** холбох.

2. Мэдрэгчүүдийн утгыг Аравтын тооллын утга болгох програмын алгоритм болон програмын код

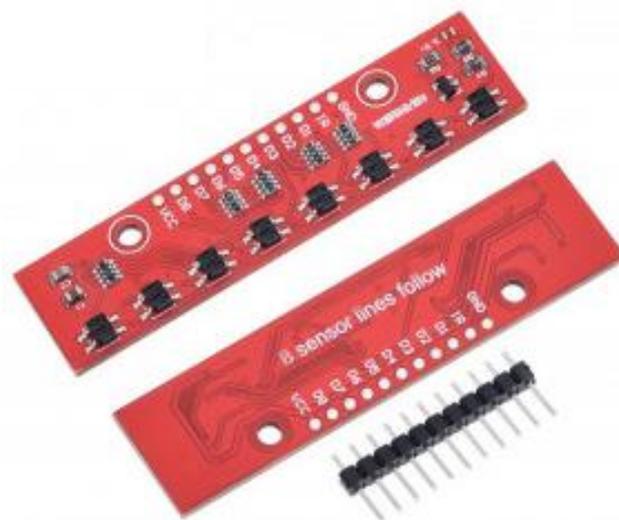




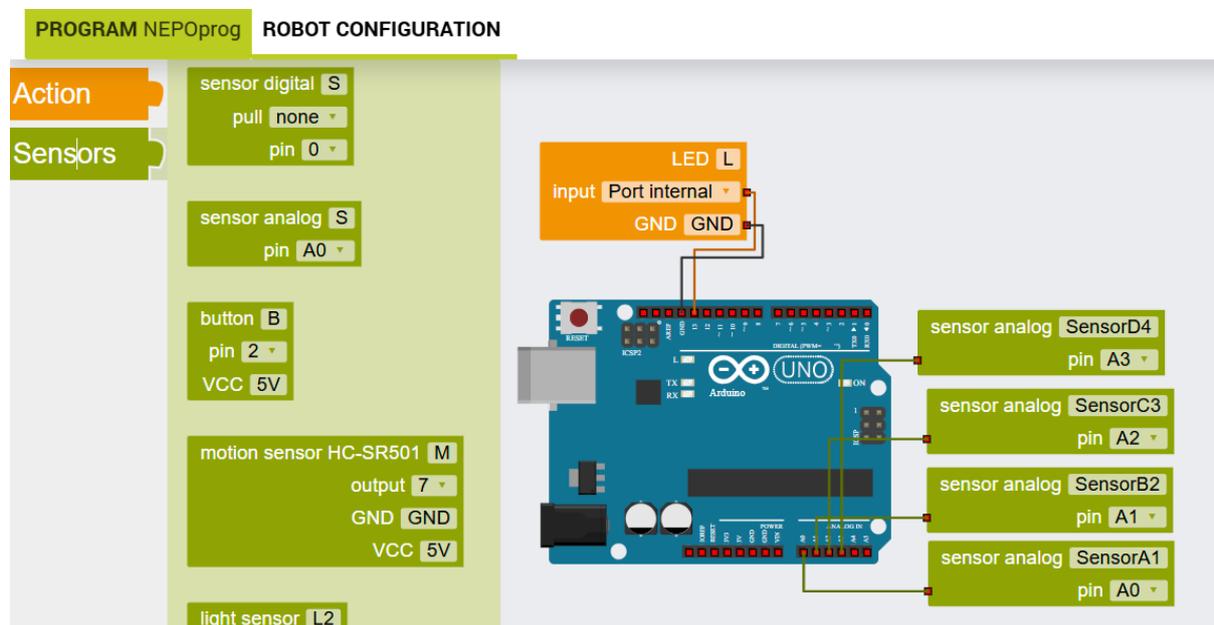
Даалгавар 2.5 . Ардуино хавтангийн A0 , A1, A2, A4 хөлнүүдэд IR мэдрэгчүүдийн 4 ширхэг аналог гаралтыг холбож, мэдрэгчүүдийн аналог утгуудыг уншиж, дэлгэцэнд хэвлэж харуулах прогармын кодыг бичээрэй.

IR мэдрэгчүүдийг Ардуино хавтанд хавтанд холбох дараалал:

- IR мэдрэгч бүрийн **GND** хөлийг Arduino хавтангийн GND хөлтэй холбох.
- IR мэдрэгч бүрийн **VCC** хөлийг Arduino хавтангийн 5V эсвэл 3.3V тэжээлтэй холбох (төхөөрөмжөөс хамаарна).
- IR мэдрэгч бүрийн **OUT** гарцыг дараах Arduino хөлүүдтэй холбох:
 - D3 гаралт → A0
 - D4 гаралт → A1
 - D5 гаралт → A2
 - D6 гаралт → A2



Мэдрэгчүүдийн аналог гаралтын утгуудыг дэлгэцэнд хэвлэж харх програмын код:



Кодны тайлбар:

- Мэдрэгчээс уншсан утга нь гэрлийн эрчмээс хамаарч өөрчлөгдөх тул ашиглах нөхцөлдөө тохируулан программ хангамжийн түвшинд босго утга тогтоох боломжтой. Жишээлбэл: `Threshold=540`
- **analogRead()** функцийг ашиглан аналог оролтоос мэдрэгчийн утгыг уншина.
- Уншсан утгуудыг Serial Monitor дээр харуулж байна.
- **delay(500)** нь 500 миллисекундын завсарлага авах бөгөөд энэ нь уншилтын давтамжийг тохируулахад ашиглагдана.

Туршилт хийх

1. Arduino хавтангаа компьютерт USB кабелиар холбоно.
2. Кодоо Arduino IDE ашиглан хавтан дээрээ оруулна.
3. Serial мониторгыг (9600 baud rate-той) нээж, мэдрэгчүүдийн өгөгдлийг шалгана.

```

+ start
- variable Sensor_A1 : Number ← 0
- variable Sensor_B2 : Number ← 0
- variable Sensor_C3 : Number ← 0
- variable Sensor_D4 : Number ← 0
- variable Threshold_A : Number ← 540
- variable Threshold_B : Number ← 540
- variable Threshold_C : Number ← 540
- variable Threshold_D : Number ← 540
- variable A : Number ← 0
- variable B : Number ← 0
- variable C : Number ← 0
- variable D : Number ← 0

repeat indefinitely
do
set Sensor_A1 to get analog value sensor SensorA1
set Sensor_B2 to get analog value sensor SensorB2
set Sensor_C3 to get analog value sensor SensorC3
set Sensor_D4 to get analog value sensor SensorD4

+ if Sensor_A1 > Threshold_A
do
set A to 0
else
set A to 1

+ if Sensor_B2 > Threshold_B
do
set B to 0
else
set B to 1

+ if Sensor_C3 > Threshold_C
do
set C to 0
else
set C to 1

+ if Sensor_D4 > Threshold_D
do
set D to 0
else
set D to 1

show on Serial Monitor " Sensor A1 "
show on Serial Monitor Sensor_A1
show on Serial Monitor A
show on Serial Monitor " Sensor B2 "
show on Serial Monitor Sensor_B2
show on Serial Monitor B
show on Serial Monitor " Sensor C3 "
show on Serial Monitor Sensor_C3
show on Serial Monitor C
show on Serial Monitor " Sensor D4 "
show on Serial Monitor Sensor_D4
show on Serial Monitor D
wait ms 500

```

```

1 // This file is automatically generated by the Arduino IDE.
2
3 #include <Arduino.h>
4
5 double __Sensor_A1;
6 double __Sensor_B2;
7 double __Sensor_C3;
8 double __Sensor_D4;
9 double __Threshold_A;
10 double __Threshold_B;
11 double __Threshold_C;
12 double __Threshold_D;
13 double __A;
14 double __B;
15 double __C;
16 double __D;
17 int _led_L = LED_BUILTIN;
18 int _input_SensorD4 = A3;
19 int _input_SensorC3 = A2;
20 int _input_SensorB2 = A1;
21 int _input_SensorA1 = A0;
22 void setup()
23 {
24     Serial.begin(9600);
25     pinMode(_led_L, OUTPUT);
26     pinMode(_input_SensorD4, INPUT);
27     pinMode(_input_SensorC3, INPUT);
28     pinMode(_input_SensorB2, INPUT);
29     pinMode(_input_SensorA1, INPUT);
30     __Sensor_A1 = 0;
31     __Sensor_B2 = 0;
32     __Sensor_C3 = 0;
33     __Sensor_D4 = 0;
34     __Threshold_A = 540;
35     __Threshold_B = 540;
36     __Threshold_C = 540;
37     __Threshold_D = 540;
38     __A = 0;
39     __B = 0;
40     __C = 0;
41     __D = 0;
42 }

```

```

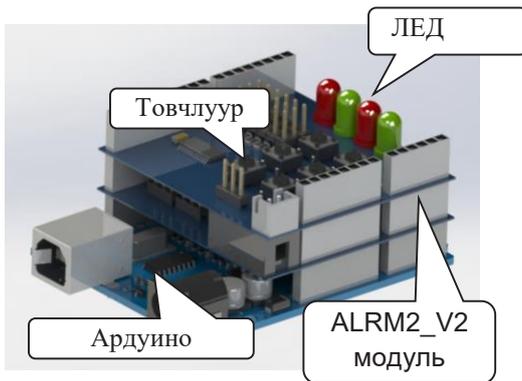
47 void loop()
48 {
49     __Sensor_A1 = analogRead(_input_SensorA1);
50     __Sensor_B2 = analogRead(_input_SensorB2);
51     __Sensor_C3 = analogRead(_input_SensorC3);
52     __Sensor_D4 = analogRead(_input_SensorD4);
53     if ( __Sensor_A1 > __Threshold_A ) {
54         __A = 0;
55     } else {
56         __A = 1;
57     }
58     if ( __Sensor_B2 > __Threshold_B ) {
59         __B = 0;
60     } else {
61         __B = 1;
62     }
63     if ( __Sensor_C3 > __Threshold_C ) {
64         __C = 0;
65     } else {
66         __C = 1;
67     }
68     if ( __Sensor_D4 > __Threshold_D ) {
69         __D = 0;
70     } else {
71         __D = 1;
72     }
73     Serial.println("Sensor A1");
74     Serial.println(__Sensor_A1);
75     Serial.println(__A);
76     Serial.println("Sensor B2");
77     Serial.println(__Sensor_B2);
78     Serial.println(__B);
79     Serial.println("Sensor C3");
80     Serial.println(__Sensor_C3);
81     Serial.println(__C);
82     Serial.println("Sensor D4");
83     Serial.println(__Sensor_D4);
84     Serial.println(__D);
85     delay(500);
86 }

```

Бүлэг 3. Ардуино хавтангийн интерфейс програмчлал



Ардуино хавтангийн интерфейс програмчлал гэдэг нь Ардуиног бусад төхөөрөмжүүдтэй холбох эсвэл хэрэглэгчидтэй харилцах интерфейстэй болгох процессыг хэлнэ. Энэ нь LCD дэлгэц, товчлуур, мэдрэгч, Bluetooth, Wi-Fi зэрэг оролт/гаралт төхөөрөмжүүдийг ашиглах тухай ойлголтыг агуулна. Бид энэхүү номдоо хамгийн түгээмэл хэрэглэгддэг интерфэйсийн програмчлалын жишээнүүдтэй танилцах болно.

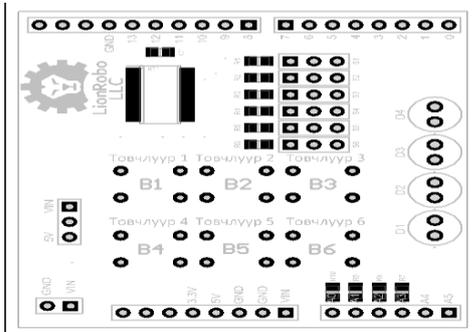


ArM2_V2 модулийн өгөгдлийн хүснэгт

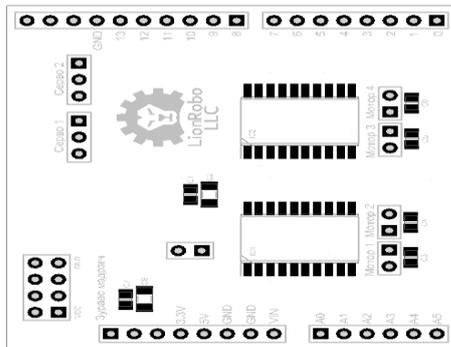
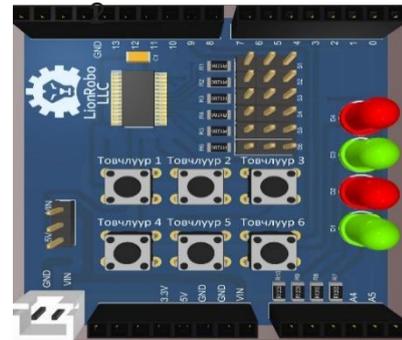
| | Оролт , гаралтын дугаар | Оролт, гаралтын утга | Үйлдэл |
|-------------------|-------------------------|-----------------------------|--|
| Гэрэл-1 | A3 | 1 буюу high | Гэрэл асах |
| | | 0 буюу low | Гэрэл унтрах |
| Гэрэл-2 | A2 | 1 буюу high | Гэрэл асах |
| | | 0 буюу low | Гэрэл унтрах |
| Гэрэл-3 | A1 | 1 буюу high | Гэрэл асах |
| | | 0 буюу low | Гэрэл унтрах |
| Гэрэл-4 | A0 | 1 буюу high | Гэрэл асах |
| | | 0 буюу low | Гэрэл унтрах |
| Товчлуур-1 | 13 | Оролт =1 бол | дарагдаагүй |
| | | Оролт =0 бол | дарагдсан |
| Товчлуур-2 | 12 | Оролт =1 бол | дарагдаагүй |
| | | Оролт =0 бол | дарагдсан |
| Товчлуур-3 | 11 | Оролт =1 бол | дарагдаагүй |
| | | Оролт =0 бол | дарагдсан |
| Товчлуур-4 | 10 | Оролт =1 бол | дарагдаагүй |
| | | Оролт =0 бол | дарагдсан |
| Товчлуур-5 | 9 | Оролт =1 бол | дарагдаагүй |
| | | Оролт =0 бол | дарагдсан |
| Товчлуур-6 | 8 | Оролт =1 бол | дарагдаагүй |
| | | Оролт =0 бол | дарагдсан |
| Серво Мотор SDA | A4 | Цуваа өгөгдлийн оролт | S1, S2, S3, S4, S5, S6 гэсэн 6 ширхэг серво моторыг удирдана |
| Серво Мотор 1 SCK | A5 | Сихрончлох давтамжийн оролт | |



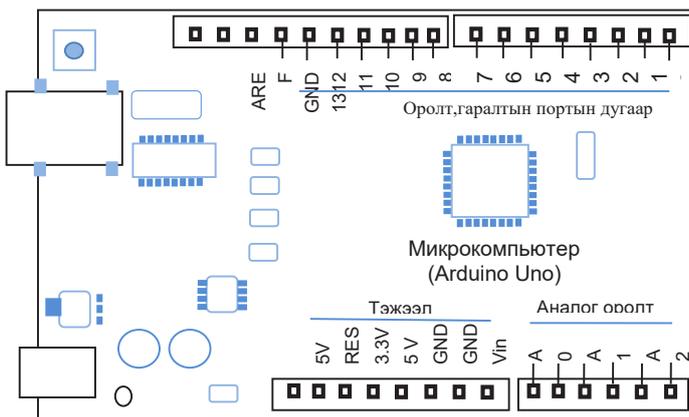
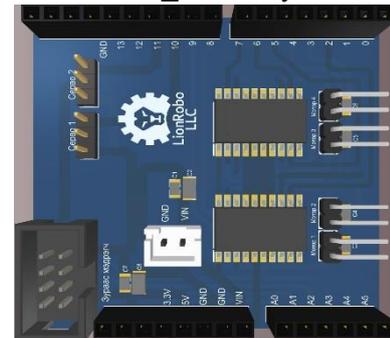
Ардиуно хавтан дээр нэмэлт модулиуд холбон илүү олон зүйлд ашиглах боломжтой байдаг. Өөрөөр хэлбэл, түүн дээр нэмж суурилуулахад зориулсан модулиудыг ашиглан Ардиуно хавтанг төрөл бүрийн зориулалтаар ашиглах боломж бүрддэг гэсэн үг юм. Зураг 10.1-д Ардиуногийн хөгжүүлэлтийн үндсэн хавтан дээр зургаан ширхэг товчлуур, дөрвөн ширхэг ЛЕД-ээс бүрдэх **ArM2_V2** Модуль 2 болон **ArM2_V2** Модуль 1 харагдаж байна.



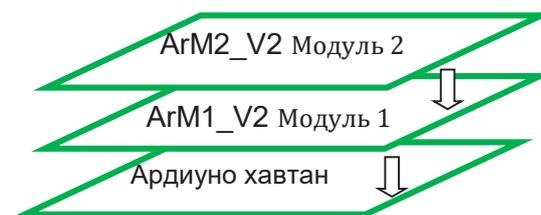
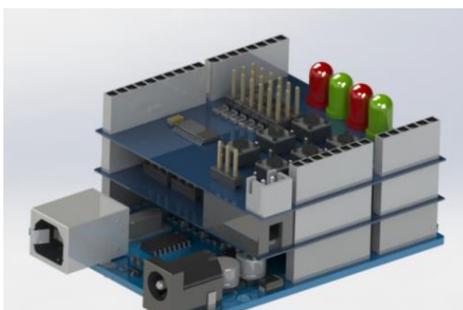
ArM2_V2 Модуль



ArM1_V2 Модуль 1



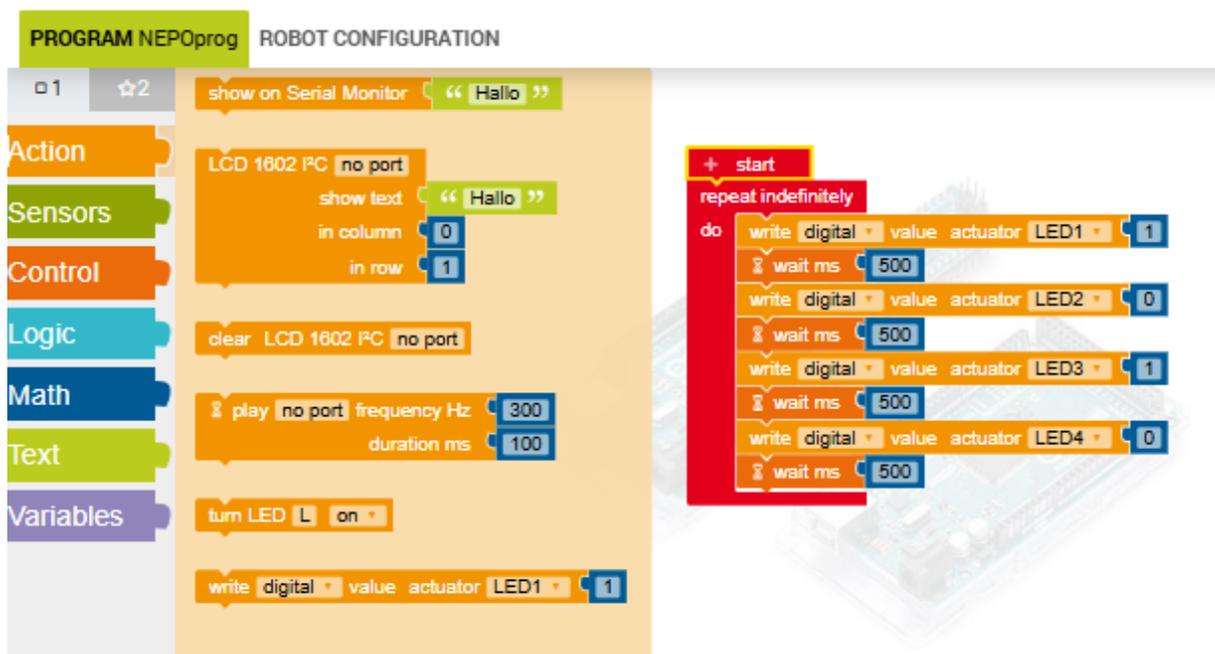
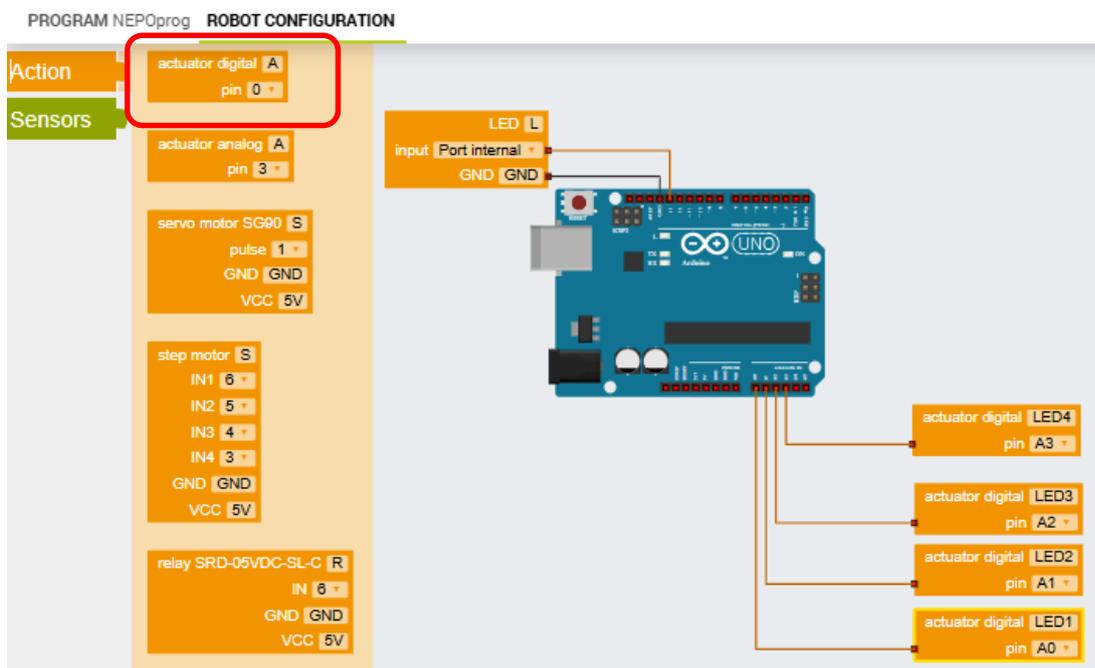
Ардиуно хавтан



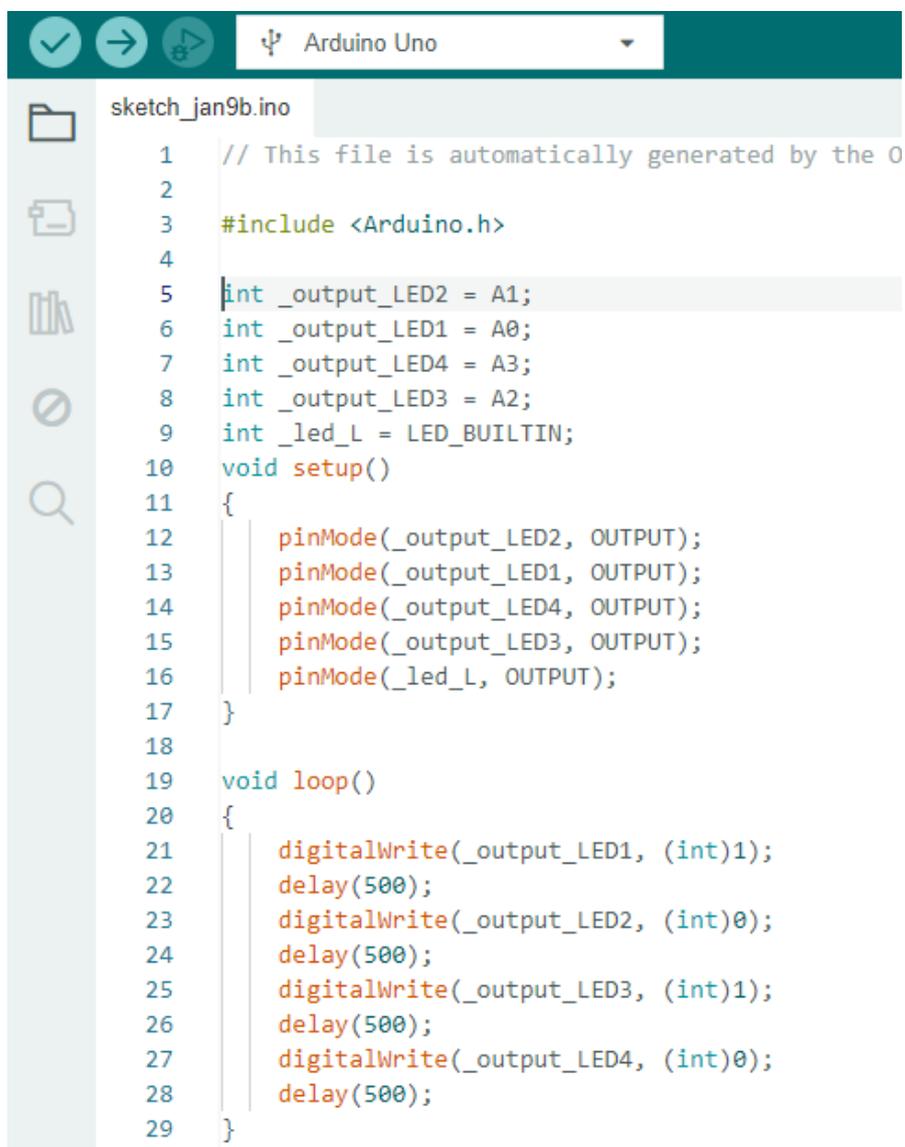
3.1 ArM2_V2 модуль 2 интерфэйсийн гэрлэн эффе́кт програмчлал

Туршилтын хавтан дээр хэлхээ угсран Ардуино хавтантай холбож ажилуулахаас гадна Ардуино хавтанд зориулагдсан нэмэлт модулуудыг ашиглах боломжтой. Ардуиногийн **ArM2_V2** модуль 2 дээрх ЛЕД гэрлүүдийг ашиглан гэрлэн эффе́кт хийх туршилтуудыг хийцгээе.

Даалгавар 3.1: Ардуино хавтангийн ArM2_V2 модуль 2-ыг ашиглан “Гэрэл 1”, “Гэрэл 2”, “Гэрэл 3”, “Гэрэл 4”-ийг 0,5 секундын хугацааны интервалтайгаар ээлжлэн асаж, унтрах програмын код бичицгээе.



Ардуино хөгжүүлэлтийн програмын цонхноос >File>New цэсийг сонгон Ардуино эскизийн програм бичих шинэ талбарыг үүсгэнэ



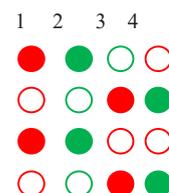
```

1 // This file is automatically generated by the 0
2
3 #include <Arduino.h>
4
5 int _output_LED2 = A1;
6 int _output_LED1 = A0;
7 int _output_LED4 = A3;
8 int _output_LED3 = A2;
9 int _led_L = LED_BUILTIN;
10 void setup()
11 {
12     pinMode(_output_LED2, OUTPUT);
13     pinMode(_output_LED1, OUTPUT);
14     pinMode(_output_LED4, OUTPUT);
15     pinMode(_output_LED3, OUTPUT);
16     pinMode(_led_L, OUTPUT);
17 }
18
19 void loop()
20 {
21     digitalWrite(_output_LED1, (int)1);
22     delay(500);
23     digitalWrite(_output_LED2, (int)0);
24     delay(500);
25     digitalWrite(_output_LED3, (int)1);
26     delay(500);
27     digitalWrite(_output_LED4, (int)0);
28     delay(500);
29 }

```

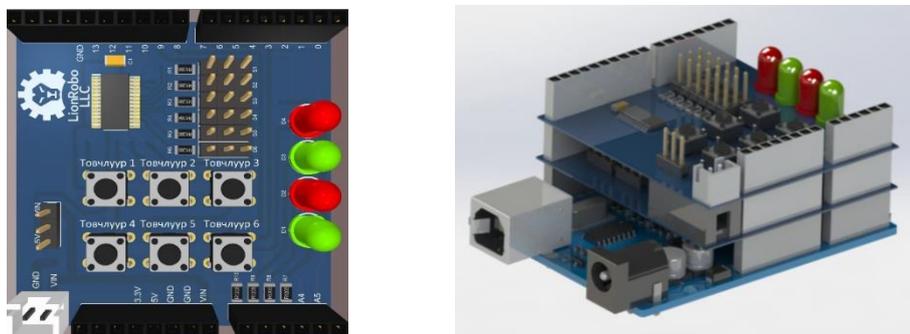
❖ **Дасгал ажил:**

- Ардуино хавтангийн “Модуль 2”-ыг ашиглан “Гэрэл-2”, “Гэрэл-3”-ыг асаах, “Гэрэл-1”, “Гэрэл-4”-ийг унтраах програмын код бич.
- Ардуино хавтангийн “Модуль 2”-ын гэрлүүдийг “Гэрэл-1 ”ба “Гэрэл-2”, “Гэрэл-3” ба “Гэрэл-4” гэсэн хослолоор анивчуулах програмын код бич. *(Гүйцэтгэх үйлдлийг зурагт харуулав)*



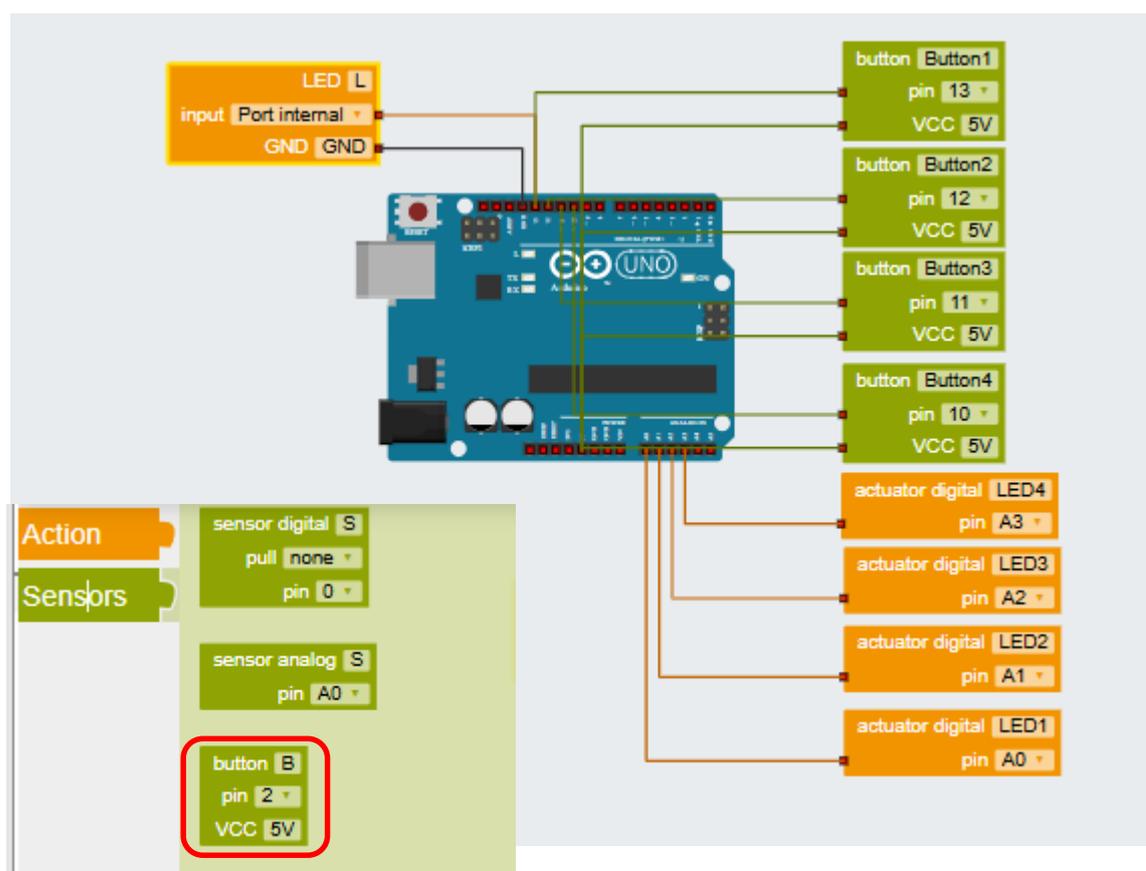
3.2 ArMV2 Модуль V2-ийн оролт, гаралтын интерфэйсийг хослуулан хэрэглэх нь

Ардуино хавтангийн ArM2_Модуль 2 интерфэйс дээрх товчлуурын өгөгдлийг цуваа портын хяналтын талбарт харахаас гадна гаралтын төхөөрөмж болох ЛЕД гэрлийн эффектээр тодорхойлж болно.



Зураг 3. 2 Ардуино хавтанг Модуль 2-той холбосон байдал

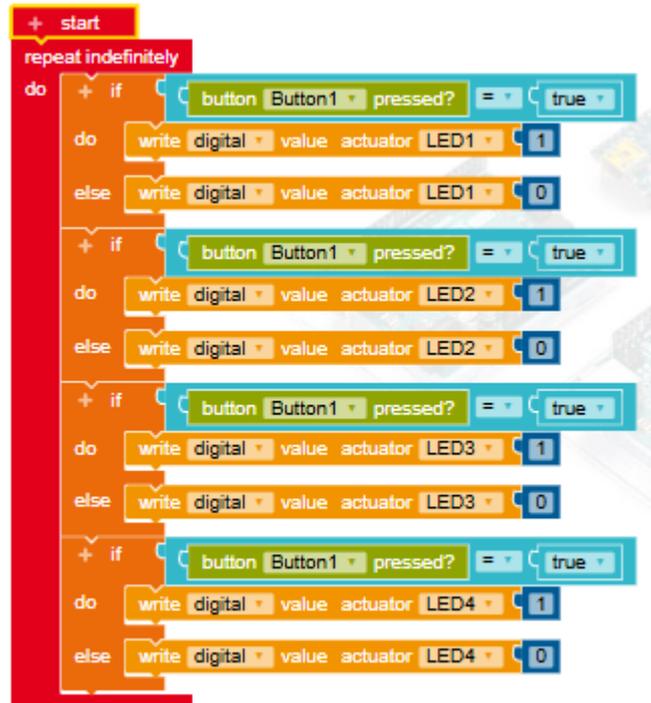
Туршилт 3.2.1: Ардуиногийн “Модуль 2”-ын “Товчлуур 1”-ийг дарахад “Гэрэл 1”, “Товчлуур 2”-ыг дарахад “Гэрэл 2”, “Товчлуур 3”-ыг дарахад “Гэрэл 3”, “Товчлуур 4”-ийг дарахад “Гэрэл 4” асаж байхаар тооцоолон програмын кодыг бичигцгээе.



```

3 #include <Arduino.h>
4
5 int _taster_Button1 = 13;
6 int _output_LED2 = A1;
7 int _output_LED1 = A0;
8 int _taster_Button4 = 10;
9 int _taster_Button3 = 11;
10 int _output_LED4 = A3;
11 int _taster_Button2 = 12;
12 int _output_LED3 = A2;
13 int _led_L = LED_BUILTIN;
14 void setup()
15 {
16     pinMode(_taster_Button1, INPUT);
17     pinMode(_output_LED2, OUTPUT);
18     pinMode(_output_LED1, OUTPUT);
19     pinMode(_taster_Button4, INPUT);
20     pinMode(_taster_Button3, INPUT);
21     pinMode(_output_LED4, OUTPUT);
22     pinMode(_taster_Button2, INPUT);
23     pinMode(_output_LED3, OUTPUT);
24     pinMode(_led_L, OUTPUT);
25 }
26
27 void loop()
28 {
29     if ( digitalRead(_taster_Button1) == true ) {
30         digitalWrite(_output_LED1, (int)1);
31     } else {
32         digitalWrite(_output_LED1, (int)0);
33     }
34     if ( digitalRead(_taster_Button1) == true ) {
35         digitalWrite(_output_LED2, (int)1);
36     } else {
37         digitalWrite(_output_LED2, (int)0);
38     }
39     if ( digitalRead(_taster_Button1) == true ) {
40         digitalWrite(_output_LED3, (int)1);
41     } else {
42         digitalWrite(_output_LED3, (int)0);
43     }
44     if ( digitalRead(_taster_Button1) == true ) {
45         digitalWrite(_output_LED4, (int)1);
46     } else {
47         digitalWrite(_output_LED4, (int)0);
48     }
49 }

```



- **Дасгал ажил:** “Модуль 2”-ын товчлуур бүрд харгалзуулан 4 өөр гэрлэн эффектийг үзүүлэх програмын код бич.(*Товчлуур бүрд харгалзах гэрлэн эффектийн дүрслэлүүдийг дараах зурагт харуулав*)

Эффект 1

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| ○ | ● | ● | ○ |
| ● | ○ | ○ | ● |
| ○ | ● | ● | ○ |
| ● | ○ | ○ | ● |

Товчлуур 1

Эффект 2

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| ○ | ● | ● | ○ |

Товчлуур 2

Эффект 3

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| ● | ○ | ○ | ○ |
| ○ | ● | ○ | ○ |
| ○ | ○ | ● | ○ |
| ○ | ○ | ○ | ● |

Товчлуур 3

Эффект 4

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| ● | ● | ● | ● |

Товчлуур 4

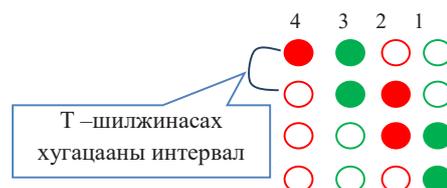
```

+ start
repeat indefinitely
do
+ if button Button1 pressed? = true
do
write digital value actuator LED1 0
write digital value actuator LED2 1
write digital value actuator LED3 1
write digital value actuator LED4 0
wait ms 200
write digital value actuator LED1 1
write digital value actuator LED2 0
write digital value actuator LED3 0
write digital value actuator LED4 1
wait ms 200
+ if button Button2 pressed? = true
do
write digital value actuator LED1 0
+ if button Button3 pressed? = true
do
write digital value actuator LED4 0
write digital value actuator LED1 1
wait ms 200
write digital value actuator LED1 0
write digital value actuator LED2 1
wait ms 200
write digital value actuator LED2 0
write digital value actuator LED3 1
wait ms 200
write digital value actuator LED3 0
write digital value actuator LED4 1
wait ms 200
+ if button Button4 pressed? = true
do
write digital value actuator LED1 1
write digital value actuator LED2 1
write digital value actuator LED3 1
write digital value actuator LED4 1

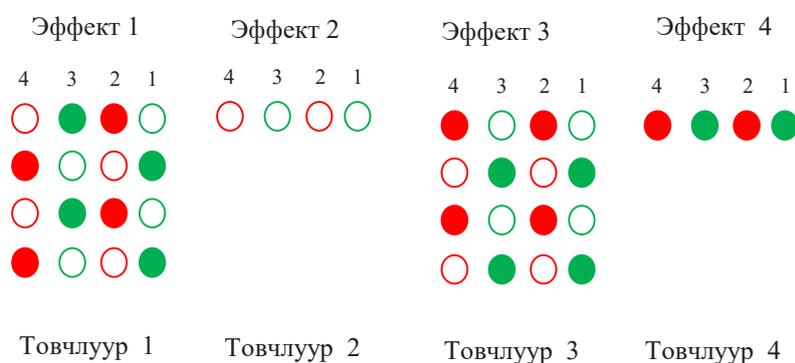
```

- “Модуль2”-ын “Товчлуур 1”-ийг дарахад “Гэрэл 4”, “Товчлуур 2”-ыг дарахад “Гэрэл 3”, “Товчлуур 3”-ыг дарахад “Гэрэл 1”, “Товчлуур 4”-ийг дарахад “Гэрэл 1” асаж, харин “Товчлуур 1” ба “Товчлуур 2”-ыг зэрэг дарахад бүх гэрлүүд унтарч, “Товчлуур 3” ба “Товчлуур 4”-ийг зэрэг дарахад бүх гэрлүүд асаж байхаар тооцоолон програмын код бич.
- “Модуль 2”-ын “Товчлуур 1”-ийг дарахад гэрлүүд ээлжлэн анивчих, “Товчлуур 2”-ыг дарахад гэрлүүд шилжин гүйж асах, “Товчлуур 3”-ыг дарахад “Гэрэл 1”, “Гэрэл 2” асаж харин “Гэрэл 3”, “Гэрэл 4” унтарч байхаар тооцоолон програмын код бич.
- “Модуль 2”-ын товчлууруудыг ашиглан түүний гэрлүүдийн шилжин асах хугацааны интервалыг өөрчлөх програмын код бич. Товчлууруудад харгалзах хугацааны утгуудыг хүснэгтээс хараарай.

| Товчлуурын төлөв | Шилжих хурд |
|---------------------------|---------------|
| Товчлуур 1-ийг дарсан үед | $T = 0.4$ сек |
| Товчлуур 2-ыг дарсан үед | $T = 0.7$ сек |
| Товчлуур 3-ыг дарсан үед | $T = 1$ сек |
| Товчлуур 4-ийг дарсан үед | $T = 1.3$ сек |

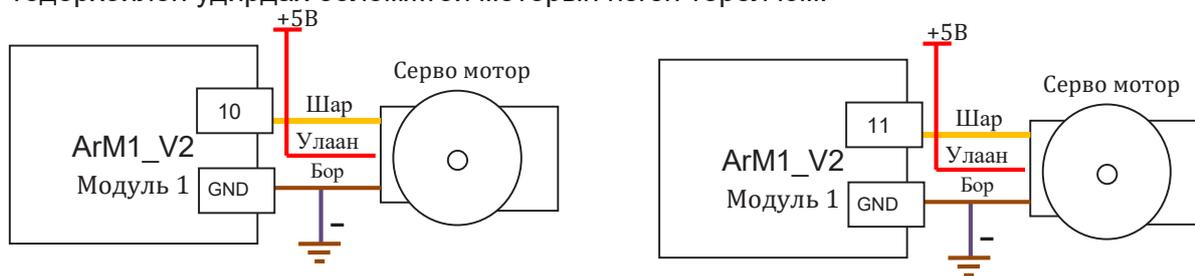


- “Модуль 2”-ын товчлуур бүрд харгалзуулан 4 өөр гэрлэн эффектийг үзүүлэх програмын код бич. (Товчлуур бүрд харгалзах гэрлэн эффектийн дүрслэлүүдийг дараах зурагт харуулав)



3.3 ArM1_V2 Модуль 1 ийн Серво моторын удирдлага

Серво мотор нь эргэлтийн байрлал, хурд, хурдатгалыг нь нарийвчлалтайгаар тодорхойлон удирдах боломжтой моторын нэгэн төрөл юм.



Зураг 3.3.1 Ардуино хавтан болон серво моторын холболт

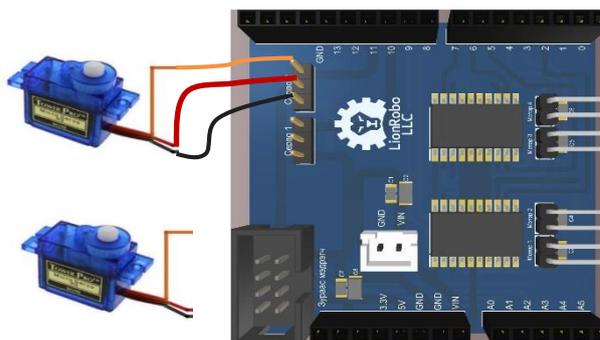
Ардуино хавтангаар серво мотор удирдахын тулд Arduino IDE-д суурилсан **Servo** санг ашиглана. Энэ сан нь серво моторыг PWM (Pulse Width Modulation) сигнал ашиглан хялбархан удирдах боломжийг олгодог.

Доорх алхмуудыг дагаж серво мотороо удирдана уу.

Серво моторын холболт

1. Серво моторын хөлнүүд:

- **VCC (Улаан утас):** Arduino хавтангийн 5V эсвэл 3.3V тэжээлд холбох.
- **GND (Хар утас):** Arduino хавтангийн GND холболтод холбох.
- **SIGNAL (Шар эсвэл Цагаан утас):** PWM гаралт дэмждэг Ардуиногийн D10 болон D11 хөлөнд холбох.



2. Тэжээлд анхаарах:

- Хэрэв серво мотор олон гүйдэл хэрэглэдэг бол гаднаас тэжээл өгөх шаардлагатай. Гадна тэжээл хэрэглэвэл, GND холболтыг Ардуино хавтангийн GND-тэй холбохоо мартуузай.

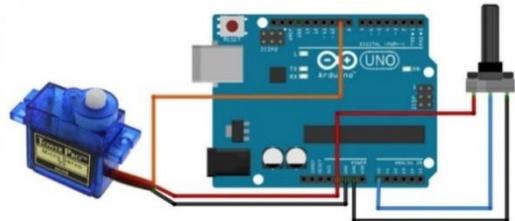
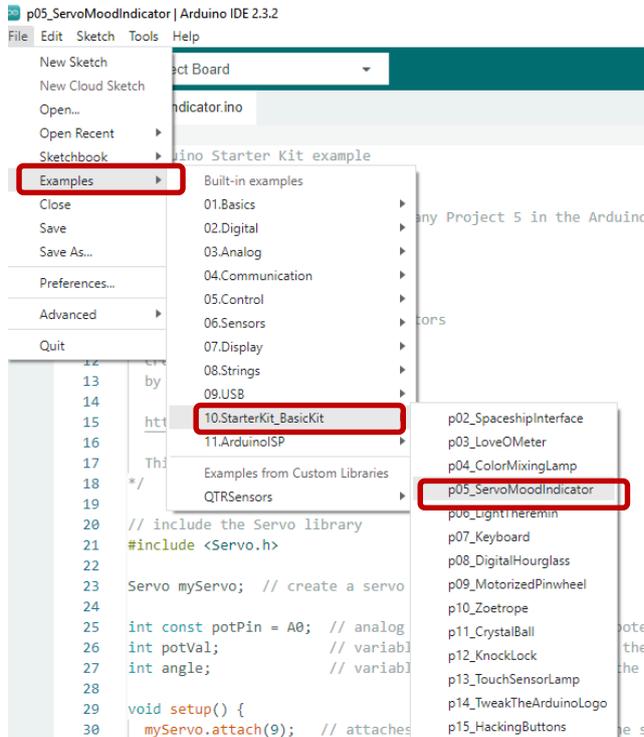
Дагалдах зөвлөмж

1. Серво моторын хэт ачаалал болон хэт халах асуудлаас сэргийлэхийн тулд тэжээлийг зөв тооцоолох.
2. Гадны тэжээл ашиглахдаа газрын (GND) холбоог Ардуино хавтантай зөв холбох.
3. Хэрэв олон серво мотор зэрэг удирдах бол гадны тэжээл ашиглахыг зөвлөж байна.

Эдгээр алхмуудыг дагаж, серво мотороо Ардуино хавтанг ашиглан амжилттай удирдаарай

Дээрх зургийн дагуу серво моторыг Ардуино хавтанд холбосны дараа Ардуино хөгжүүлэлтийн санд байгаа *ServoMoodIndicator* нэртэй программыг ашиглан моторыг эргүүлэх туршилтыг хийцгээе.

Туршилт 3.3. Доорх зургийн дагуу серво мотор болон потенциотметрийг Ардуино хавтанд холбосны дараа Ардуино хөгжүүлэлтийн санд дахь *ServoMoodIndicator* нэртэй программыг ашиглан моторыг потенциотметрийн эргэлтээр удирдах туршилтыг хийцгээе.



Зураг 3.3.3 Ардуино хавтан, серво мотор болон потенциотметрын холболт

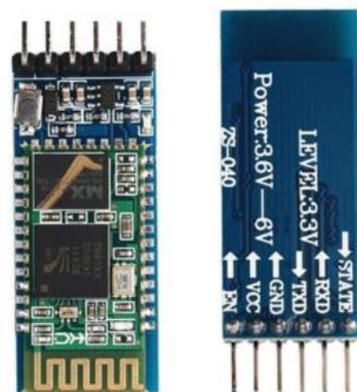
```

33
34 void loop() {
35   potVal = analogRead(potPin); // read the value
36   // print out the value to the Serial Monitor
37   Serial.print("potVal: ");
38   Serial.print(potVal);
39
40   // scale the numbers from the pot
41   angle = map(potVal, 0, 1023, 0, 179);
42
43   // print out the angle for the servo motor
44   Serial.print(", angle: ");
45   Serial.println(angle);
46
47   // set the servo position
48   myServo.write(angle);
49
50   // wait for the servo to get there
51   delay(15);
52 }
53

```

3.4 Bluetooth модуль ашиглан Ардуиног удирдах (HC-05):

Ардуиног ашиглан Bluetooth модулийг холбож, удирдах системийг үүсгэх нь маш сонирхолтой бөгөөд практик төслийн жишээ юм. Ихэвчлэн **HC-05** эсвэл **HC-06** Bluetooth модуль ашигладаг бөгөөд эдгээрийг Ардуинотой хялбархан холбож болно



1. HC-05/HC-06 Bluetooth модулийн холболт

- **VCC** → Ардуино-ийн **5V**
 - **GND** → Ардуино-ийн **GND**
 - **TXD** → Ардуино-ийн **RX (10-р пин)**
(програмчлагдах RX)
 - **RXD** → Ардуино-ийн **TX (11-р пин)**
(програмчлагдах TX)
- ⚠ **RXD ба TXD хооронд 1kΩ ба 2kΩ резистор ашиглан түвшин бууруулах шаардлагатай байж болно.**

Туршилт 3.4. Bluetooth-оос өгөгдөл хүлээн авч, LED асаах/унтраах жишээ код:

```

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX = 10, TX = 11

void setup() {
  Serial.begin(9600);           // Serial монитор
  BTSerial.begin(9600);        // Bluetooth-ийн хурд
  Serial.println("Bluetooth Ready!");
}

void loop() {
  if (BTSerial.available()) {
    char data = BTSerial.read(); // Bluetooth-оос өгөгдөл унших
    Serial.print("Received: ");
    Serial.println(data);

    if (data == '1') {
      Serial.println("LED ON");
    } else if (data == '0') {
      Serial.println("LED OFF");
    }
  }
}

```

Гаралтыг шалгах

1. **Bluetooth төхөөрөмж холбох:**
 - Гар утас эсвэл компьютер ашиглан Bluetooth-оо асаана.
 - HC-05 эсвэл HC-06 Bluetooth модулийг хайж, холбогдоно.
 - **PIN код:** Ихэнхдээ **1234** эсвэл **0000** байна.
2. **Bluetooth терминал апп ашиглах:**
 - Android-д зориулсан "Serial Bluetooth Terminal" эсвэл ижил төрлийн апп суулгана.
 - HC-05/HC-06-тай холбогдож, командыг илгээж эхэлнэ.
3. **Команд илгээх:**
 - **1** илгээхэд LED асна.
 - **0** илгээхэд LED унтарна.

Кодны тайлбар

1. **SoftwareSerial:** Arduino-ийн RX, TX пинүүдийг програмчлагдах пинүүдэд (D10, D11 гэх мэт) тохируулна.
2. **Команд шалгах (if нөхцөл):** Илгээсэн өгөгдлийг шалган тухайн функцыг гүйцэтгэнэ.
3. **Bluetooth терминал ашиглах:** Удирдлагын командыг терминалаас илгээж болно.

3.5 Хэт авианы мэдрэгч (HC-SR04) ашиглан зайг хэмжих:

HC-SR04 нь хэт авианы мэдрэгч бөгөөд ойролцоо байгаа биетийн зайг хэмжихэд ашиглагддаг. Энэ нь **Trig** (Trigger), **Echo**, **VCC**, **GND** гэсэн дөрвөн холболттой. Зайг хэмжихэд дууны долгион ашигладаг тул маш нарийвчлалтай бөгөөд зайг мм-ээс м хүртэл хэмжих боломжтой.



HC-SR04 модуль хэрхэн ажилладаг вэ?

1. **Trigger пин (TRIG):** Богино хугацааны (10 микросекунд) HIGH сигнал илгээж дууны долгион үүсгэнэ.
2. **Echo пин (ECHO):** Объектээс ойж ирсэн дууны долгионыг хүлээн авч HIGH сигналын хугацааг хэмжинэ.
3. **Хугацаа ба зайны харилцаа:** HIGH сигналын үргэлжлэх хугацааг ашиглан дууны хурдыг тооцож, объект хүртэлх зайг тодорхойлно.

Томьёолол: $\text{Зай} = (\text{Цагхугацаа} \times \text{Дууныхурд}) / 2$

Ду HC-SR04 мэдрэгчийн холболт

1. **VCC:** Arduino-ийн 5V-д холбох.
2. **GND:** Arduino-ийн GND-д холбох.
3. **TRIG:** Arduino-ийн дижитал гаралт (жишээ нь, 9-р хөл).
4. **ECHO:** Arduino-ийн дижитал оролт (жишээ нь, 10-р хөл).

Техникийн үзүүлэлт

| Үзүүлэлт | Утга |
|----------------|---------------|
| Хүчдэл | 5V DC |
| Гүйдэл | 15 mA |
| Илрүүлэх зай | 2 см - 400 см |
| Нарийвчлал | ±3 мм |
| Илрүүлэх өнцөг | 15 градус |

Програмын тайлбар

- **digitalWrite(trigPin, HIGH):** TRIG пинд богино хугацааны (10 микросекунд) HIGH сигнал илгээнэ.
- **pulseIn(echoPin, HIGH):** ECHO пинд ирж буй HIGH сигналын хугацааг микросекундээр хэмжинэ.
- Зайг **тооцоолох:** Утгыг см болгон хөрвүүлэхийн тулд **0.0343**-ыг үржүүлж, **2**-т хуваана (явах ба буцах замыг тооцдог тул).

Зөвлөмж

- Зайг зөв хэмжихийн тулд мэдрэгчийг объект руу чиглүүлнэ.
- HC-SR04 мэдрэгчийг өөр тэжээлтэй төхөөрөмжид ойртуулж ашиглахдаа цахилгаан сөрөг нөлөөнөөс зайлсхийхийн тулд тусгаарлах.
- Гадны орчинд (салхи, дуу чимээ гэх мэт) мэдрэгчийн нарийвчлал бага зэрэг алдагдаж болзошгүй.

Туршилт 3.5. Доорх кодыг ашиглан HC-SR04 мэдрэгчээр зай хэмжих боломжтой:

```
const int trigPin = 9; //
TRIG холболт const int echoPin = 10; // ECHO холболт
void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600); // Serial Monitor ашиглах
}
void loop()
{
  // TRIG пинд богино хугацааны HIGH сигнал илгээх
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW); // ECHO пинд HIGH сигналын үргэлжлэх хугацааг хэмжих long
  duration = pulseIn(echoPin, HIGH); // Зайг тооцоолох (см)
  float distance = (duration * 0.0343) / 2; // Serial Monitor дээр зайг харуулах S
  Serial.print("Distance: ");
  Serial.print(distance); Serial.println(" cm");
  delay(500); // 500ms хүлээх
}
```

3.6 Чийг мэдрэгч (FC-28) модулийг ашиглан хөрсний чийгшлийг тодорхойлох

Чийг мэдрэгч (FC-28) модулийг ашиглан хөрсний чийгийн түвшинг хэмжих нь хялбар бөгөөд хэрэгцээтэй шийдэл юм. Энэ мэдрэгч нь аналог болон тоон гаралтаар ажиллах боломжтой бөгөөд хөрсөн дэх чийгийн хэмжээ ихэсэх тусам гаралтын утга буурдаг.

FC-28 мэдрэгчийн үндсэн хэсгүүд

- **VCC** – Модульд хүчдэл өгөх (3.3V эсвэл 5V).
- **GND** – Газардуулга.
- **A0 (Аналог гаралт)** – Чийгийн утгыг аналог дохиогоор дамжуулна.

Модулийн потенциометроор дижитал гаралтын босго утгыг тохируулна.

- **VCC** → Arduino-ийн 5V.
- **GND** → Arduino-ийн GND.
- **A0** → Arduino-ийн A0 пин (аналог утга унших).
- **D0** (хэрэглэх бол) → Arduino-ийн тоон оролт пин.

❖ Програмчлал

Доорх кодыг ашиглан чийгийн утгыг хэмжиж, Serial Monitor дээр харуулах боломжтой



Туршилт 3.6.1 Аналог гаралтыг ашигласан програмын код:

```
const int sensorPin = A0; // Аналог оролт(A0)
int sensorValue; // Мэдрэгчийн уншсан утга (0-1023)
void setup() {
  Serial.begin(9600); // Serial Monitor ашиглах
}
void loop() {
  sensorValue = analog
  Read(sensorPin); // Мэдрэгчийн утгыг унших (0-1023)
  // Хөрсний чийгийн түвшинг хувиар харуулах
  float moisturePercent = map(sensorValue, 1023, 0, 0, 100); // 0% (хуурай) - 100% (нойтон)
  Serial.print("Soil Moisture: ");
  Serial.print(moisturePercent);
  Serial.println("%"); delay(1000);
  // 1 секунд хүлээх
}
```

Туршилт 3.6.2 Тоон гаралтыг ашигласан програмын код:

```
const int digitalPin = 2; // D0 холбогдсон хөл
const int ledPin = 13; // LED холбогдсон хөл
void setup() {
  pinMode(digitalPin, INPUT); // Дижитал оролтыг тохируулах
  pinMode(ledPin, OUTPUT); // LED-г гаралт болгох
  Serial.begin(9600);
}
void loop() {
  int moistureStatus = digitalRead(digitalPin); // Дижитал утгыг унших (HIGH/LOW)
  if (moistureStatus == LOW) { // Хөрсний чийг их байвал
    digitalWrite(ledPin, HIGH); // LED асаах
    Serial.println("Soil is Wet");
  } else { // Хөрс хуурай бол
    digitalWrite(ledPin, LOW); // LED унтраах
    Serial.println("Soil is Dry");
  }
  delay(1000); // 1 секунд хүлээх
}
```

❖ Програмын тайлбар:

- **analogRead(sensorPin):** Чийгийн утгыг 0-1023 хооронд уншина.
 - **0:** Хамгийн их чийгтэй (нойтон).
 - **1023:** Хамгийн бага чийгтэй (хуурай).
- **map(sensorValue, 1023, 0, 0, 100):** Уншсан утгыг хувиар хөрвүүлнэ.
- **digitalRead(digitalPin):** Дижитал гаралтын HIGH/LOW байдлыг шалгана.
- **LED ашиглах:** Хөрсний чийгийн төлөвийг харах нэмэлт туслах хэрэгсэл.

❖ Нэмэлт: Усалгааны системтэй холбох

Автоматаар услах систем хийхийг хүсвэл дараах зүйлсийг нэмнэ:

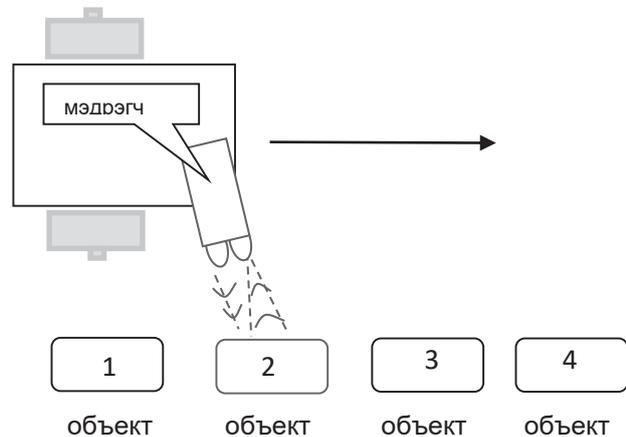
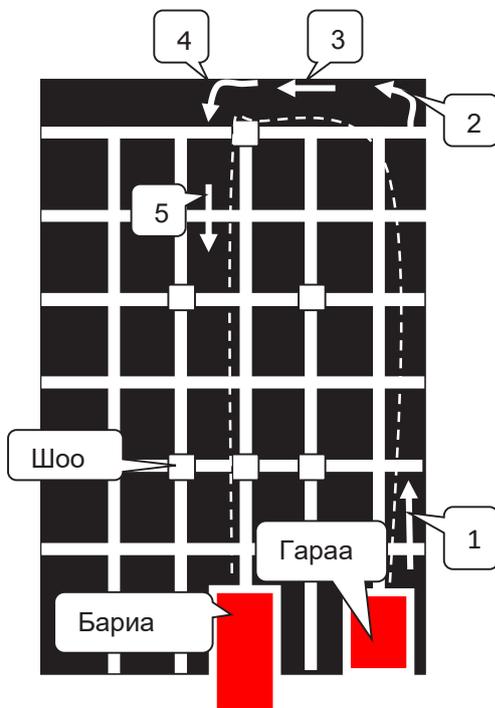
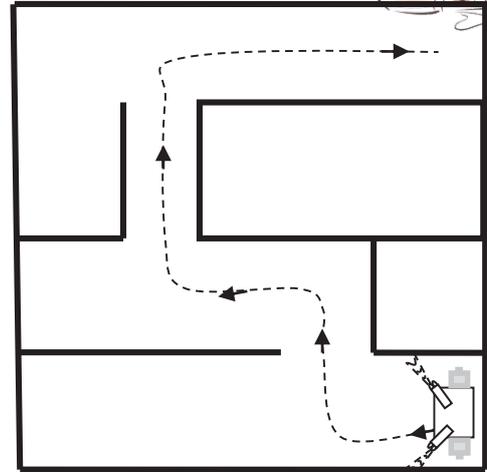
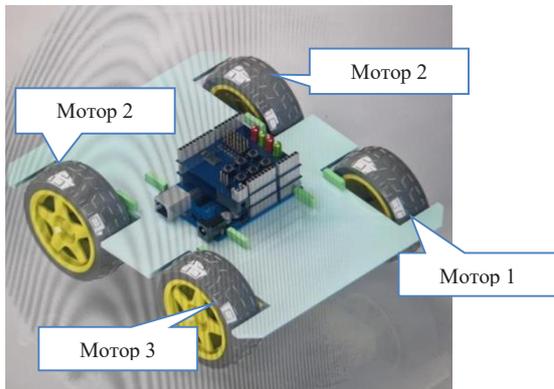
- **Реле модуль (Relay Module):** Насос эсвэл усалгааны төхөөрөмжийг

- **Кодны өөрчлөлт:**
 - Хөрс хуурай (чийг бага) үед реле асаах.
 - Хөрс хангалттай чийгтэй үед реле унтраах.
- ❖ **Зөвлөмж**
- **Мэдрэгчийг удаан ашиглах:**
Мэдрэгчийг удаан хугацаанд хөрсөнд байлгавал зэврэх магадлалтай тул богино хугацааны хэмжилтэд ашиглахыг зөвлөж байна.
- **Бодит хэрэглээ:**
Усалгааны автомат системд ашиглахдаа гадны реле модуль ашиглах нь зүйтэй. Насос эсвэл мотор холбох үед аюулгүй байдлыг хангах.
- **Хөрсний онцлог:**
Мэдрэгчийн үр дүн хөрсний төрлөөс шалтгаалж өөрчлөгдөж болно. Иймд бодит нөхцөлд тохируулга хийх хэрэгтэй.

Модулийн техникийн үзүүлэлт

| Үзүүлэлт | Утга |
|----------------|--------------------|
| Ажиллах хүчдэл | 3.3V - 5V |
| Оролтын гүйдэл | 35 mA |
| Аналог утга | 0 - 1023 |
| Дижитал гаралт | HIGH/LOW |
| Мэдрэх хүрээ | Хуурайгаас чийгтэй |

Бүлэг 4. Ухаалаг роботын програмчлал



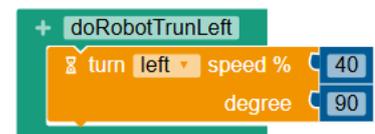
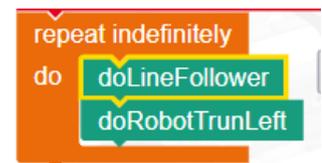
Компьютерийн хэлнүүдийн **бүтэц, зохион байгуулалт** нь өөр өөр байдаг. Харин компьютерийн бүх хэлнүүдийн ажиллах зарчим нь нэг ижил **логик** дээр үндэслэдэг. Иймээс, компьютерийн бүх програмууд ижил төстэй байдлаар ажилладаг бөгөөд **дараалал** гэсэн програмчлалын үндсэн **логикийг** баримталдаг.

Үндсэн програм гэж юу вэ?

Үндсэн програм гэдэг нь програмын эхлэх болон дуусах үйлдлийг тодорхойлох програмын гол хэсэг юм.

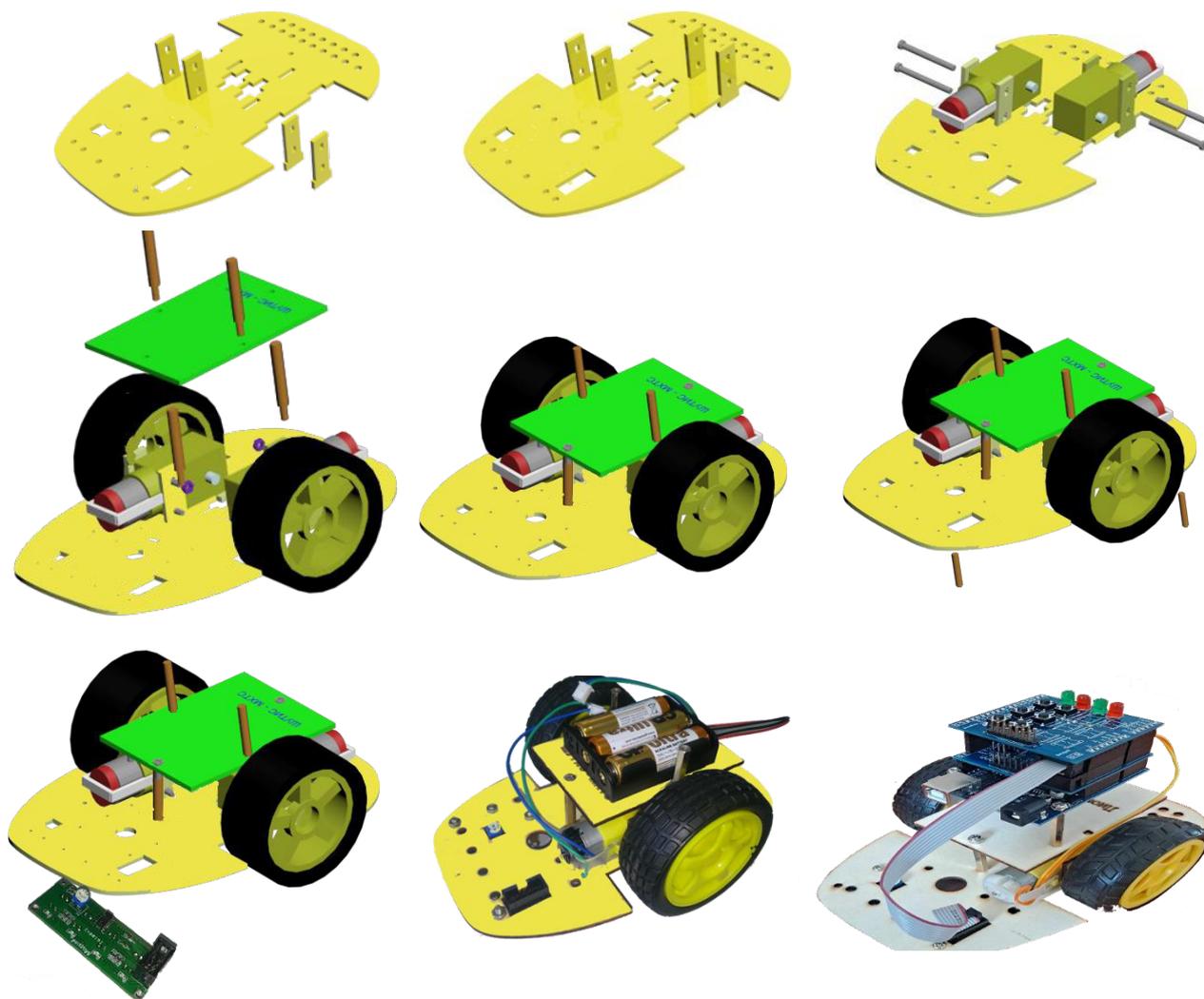
Туслах програм гэж юу вэ?

Туслах програм гэдэг нь үндсэн програмаас тусдаа үйлдэл гүйцэтгэх чадвар бүхий нэмэлт програмын хэсэг юм. **Туслах програм** үндсэн програм дотор програмын нэгэн хэсэг болон ажиллаж болно. Өөрөөр хэлбэл, үндсэн програмаас тухайн дэд програмын нэрийг дуудах үед програмын ажиллагаа дэд програмын хэсэгт шилжинэ. Програмын ажиллагаа нь дэд програм дахь блокуудын үйлдлийг дуусгаад үндсэн програм лээр буцаж очлог

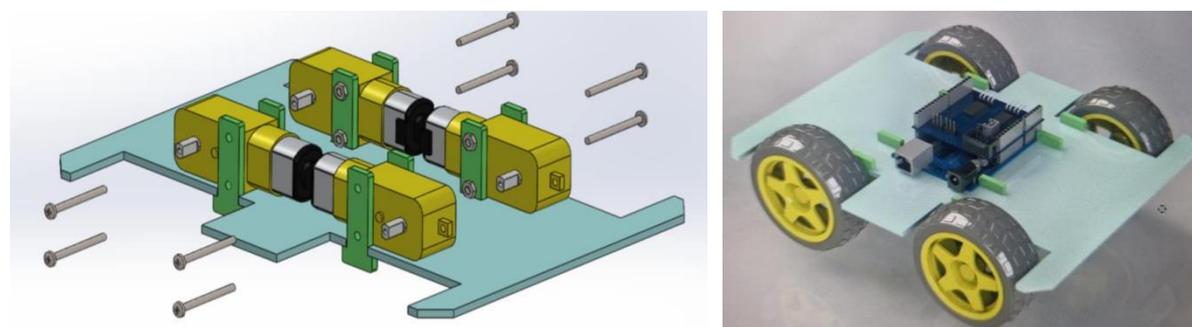


4.1 Роботын загварчлал

❖ Хоёр дугуйтай роботын загварчлал



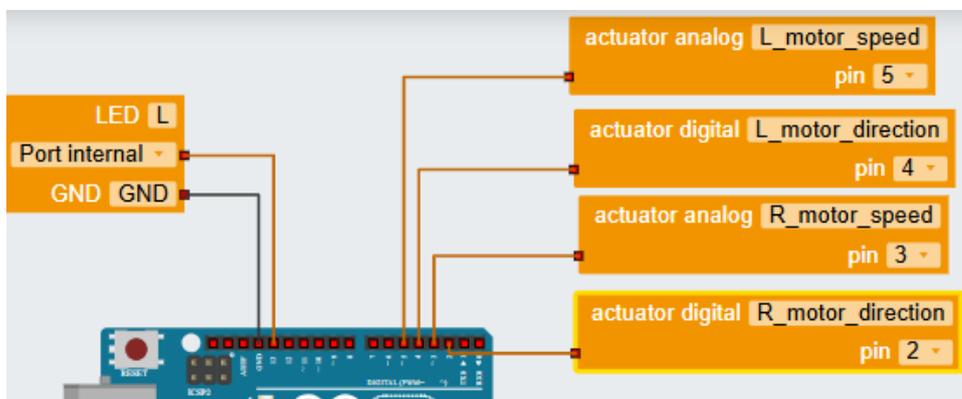
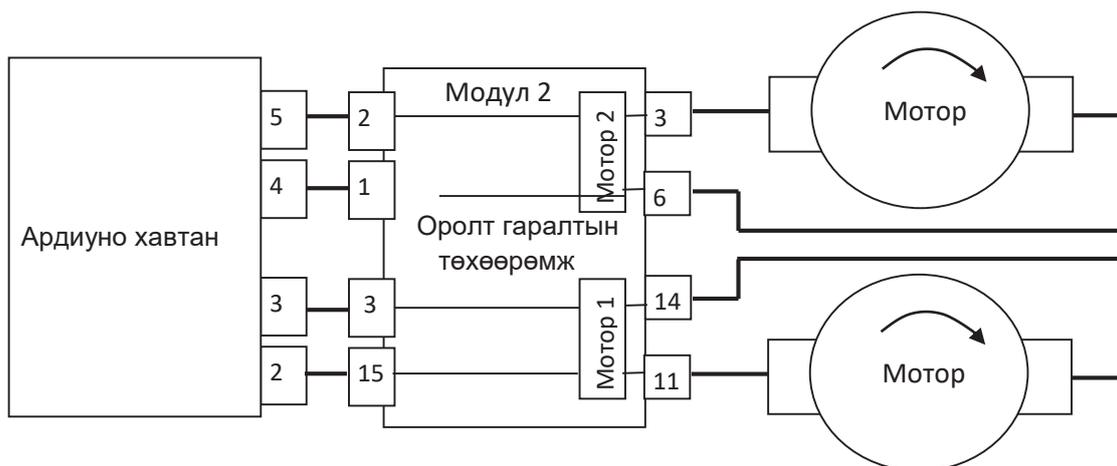
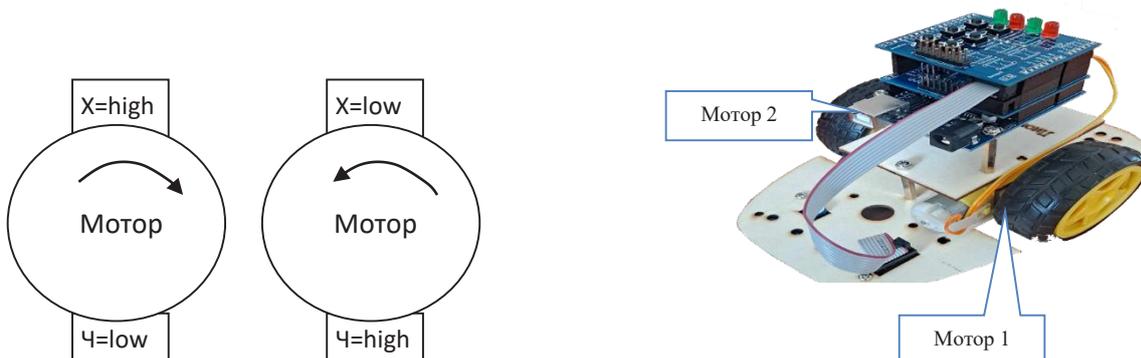
❖ Дөрвөн дугуйтай роботын загварчлал



4.2 Моторын удирдлагын програмчлал

Хоёр дугуйтай роботын удирдлага

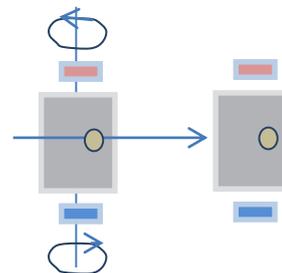
| Роботын моторын хурдныг удирдах утгууд | | | | | | |
|--|-----------|-----------|-----------|-------------|---------|-----------|
| | доод хурд | бага хурд | дунд хурд | Дундаж хурд | Их хурд | Дээд хурд |
| Хурдны удирдлага | 70 | 71-99 | 100-149 | 150-199 | 200-254 | 255 |



Дасгал 4.2.1 Роботын мотор-1 болон мотор-2 –ыг эсрэг чиглэлд эргүүлэн роботыг урагшаа явуулаад 2 секундын дараа зогсоох програмын кодыг бич.

```

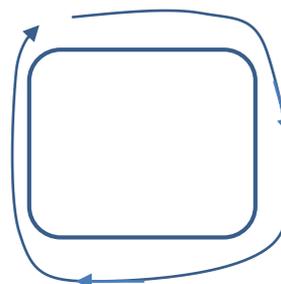
+ start
repeat indefinitely
do
  write digital value actuator L_motor_direction 0
  write analog value actuator L_motor_speed 200
  write digital value actuator R_motor_direction 0
  write analog value actuator R_motor_speed 200
  wait ms 200
  write analog value actuator L_motor_speed 0
  write analog value actuator R_motor_speed 0
  
```



Дасгал 4.2.2 Зурагт өгөгдсөн замчлалаар цагийн зүүний дагуу роботыг эргэлдүүлэн удирдах програмын кодыг бич.

```

+ start
repeat indefinitely
do
  write digital value actuator L_motor_direction 0
  write analog value actuator L_motor_speed 200
  write digital value actuator R_motor_direction 0
  write analog value actuator R_motor_speed 200
  wait ms 1000
  write analog value actuator R_motor_speed 0
  wait ms 500
  
```



Даалгавар 4.2. Моторын удирдлагын програмчлал

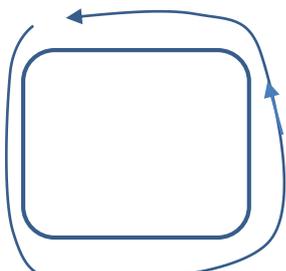
Даалгавар 4.2.1 Роботын мотор-1 болон мотор-2 –ыг ижил чиглэлд эргүүлэн роботыг тэнхлэг дээр эргэлдүүлэх програмын кодыг бич.

Даалгавар 4.2.2 Роботын тэнхлэг дээр нь 1 секунд баруун тийш эргэлдүүлээд дараа нь хойш явуулах програмын кодыг бич.

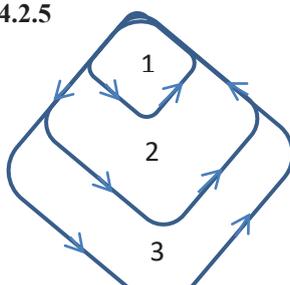
Даалгавар 4.2.3 Роботын тэнхлэг дээр нь 0.5 секунд баруун тийш эргэлдүүлээд дараа нь 0.1 секунд зогсоогоод 0.5 секунд зүүн тийш эргэлдүүлэх програмын кодыг бич.

Даалгавар 4.2.4-4.2.7 Зурагт өгөгдсөн замчлалын дагуу цагийн зүүний эсрэг роботыг эргэлдүүлэн удирдах програмын кодыг бич.

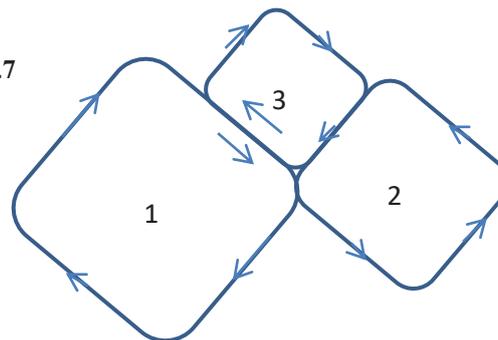
4.2.4



4.2.5



4.2.7



4.3. Bluetooth модуль ашигласан алсын удирдлагатай роботын програмчлал

```
#define A_1A 3
#define A_1B 2
#define B_1A 5
#define B_1B 4

#define MOTOR_SPEED 150

unsigned char A_speed=0, B_speed=0;
boolean A_dir=0, B_dir=0;
void setup()
{ pinMode(A_1A, OUTPUT);
  pinMode(A_1B, OUTPUT);
  pinMode(B_1A, OUTPUT);
  pinMode(B_1B, OUTPUT);

  Serial.begin(9600);
  Serial.println("RC Car is ready !");}

void loop()
{ unsigned char key_in=0;

  if(Serial.available() ){
    key_in=Serial.read();

    Serial.println(key_in);
    motor_control(key_in);
    motor_drive();
  }

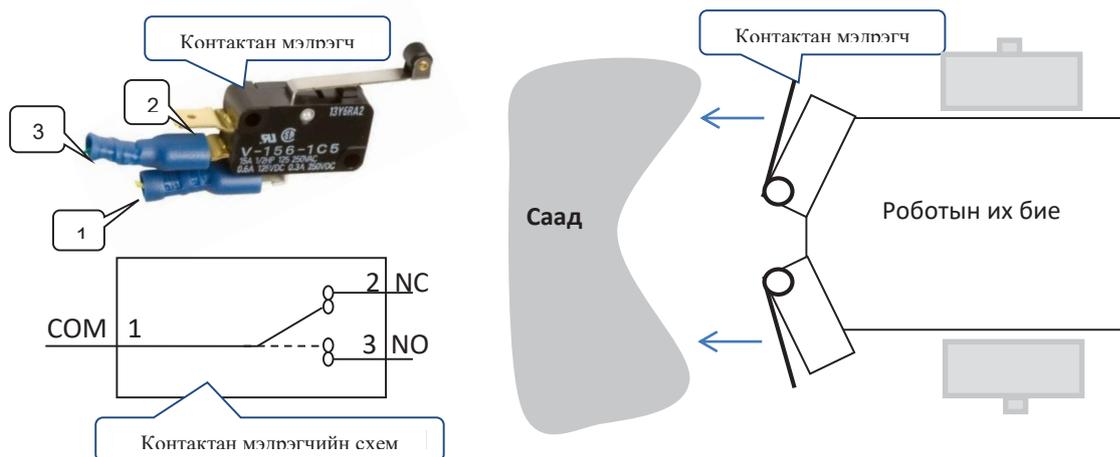
  void motor_control(unsigned char key_in)
  {
    if(key_in==76){
      Serial.println("LEFT");
      Serial.println(key_in);
      A_dir=1;
      B_dir=1;
      A_speed=MOTOR_SPEED;
      B_speed=MOTOR_SPEED; }

    else if(key_in==70){
      Serial.println("FORWARD");
      Serial.println(key_in);
      A_dir=0;
      B_dir=1;
      A_speed=MOTOR_SPEED;
      B_speed=MOTOR_SPEED; }
    else if(key_in==66) {
      Serial.println("BACK");
      Serial.println(key_in);
      A_dir=1;
      B_dir=0;
      A_speed=MOTOR_SPEED;
      B_speed=MOTOR_SPEED; }
```

```
if(key_in==82){
  Serial.println("RIGTH");
  Serial.println(key_in);
  A_dir=0;
  B_dir=0;
  A_speed=MOTOR_SPEED;
  B_speed=MOTOR_SPEED;
}
else if(key_in==76){
  Serial.println("LEFT");
  Serial.println(key_in);
  A_dir=1;
  B_dir=1;
  A_speed=MOTOR_SPEED;
  B_speed=MOTOR_SPEED;
}
else if(key_in==83){
  Serial.println("STOP");
  Serial.println(key_in);
  A_dir=0;
  B_dir=0;
  A_speed=0;
  B_speed=0;
}
}
}

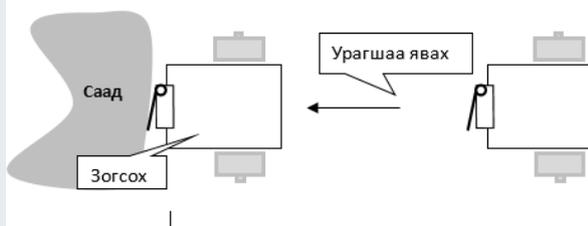
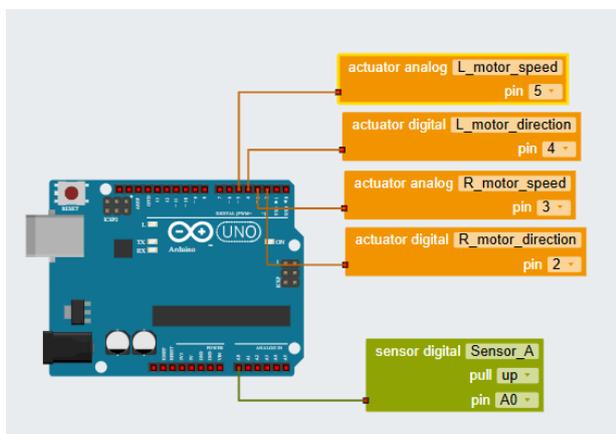
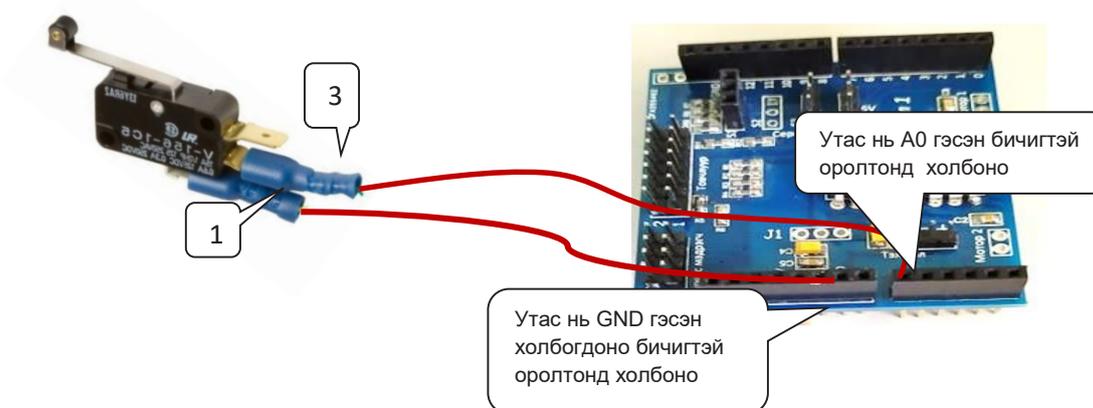
void motor_drive()
{
  if(A_dir==0){
    digitalWrite(A_1A, LOW);
    analogWrite(A_1B, A_speed);
  }
  else{
    analogWrite(A_1A, A_speed);
    digitalWrite(A_1B, LOW);
  }
  if(B_dir==1) {
    digitalWrite(B_1A, LOW);
    analogWrite(B_1B, B_speed);
  }
  else{
    analogWrite(B_1A, B_speed);
    digitalWrite(B_1B, LOW);
  }
}
```

4.4 Контактан мэдрэгч бүхий роботын програмчлал

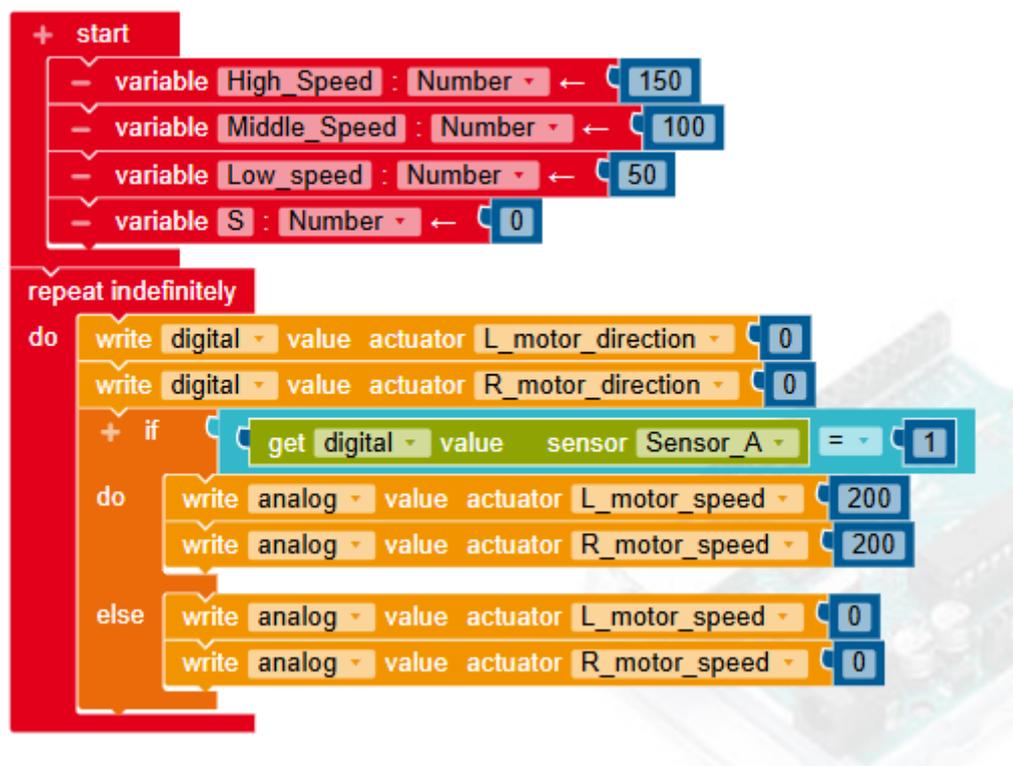


Дасгал 4.4.1 Доорх дарааллын дагуу гүйцэтгэлийг хийж, роботын ажиллагааг туршаарай.

- Роботын урд талд тохиромжтой байрлалыг сонгон контактан мэдрэгчийг байрлуул.
- Доорх зургийн дагуу контактан мэдрэгчийг Ардуиногийн модул 2- т холбо.
- Робот нь тэжээл залгах үед урагшаа явж эхлэх бөгөөд роботын өмнө объект тааралдах үед контактан мэдрэгчээр саадыг илрүүлж, роботыг зогсоох програмын кодыг бич.

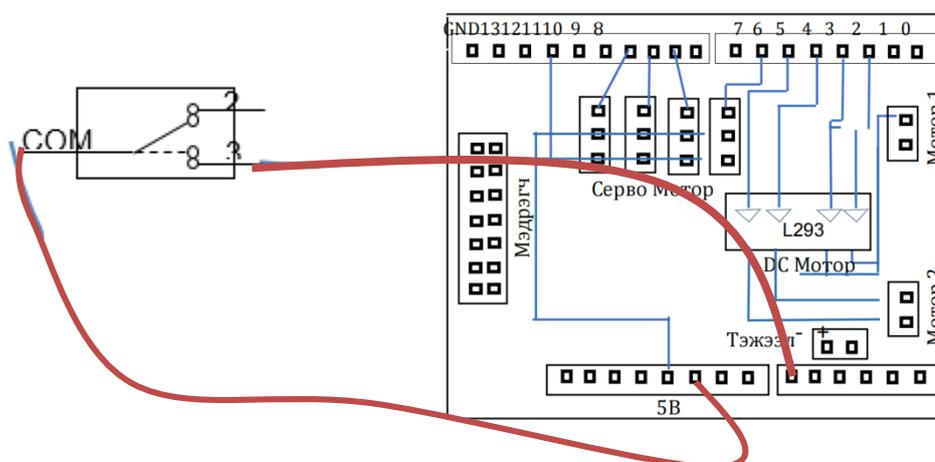


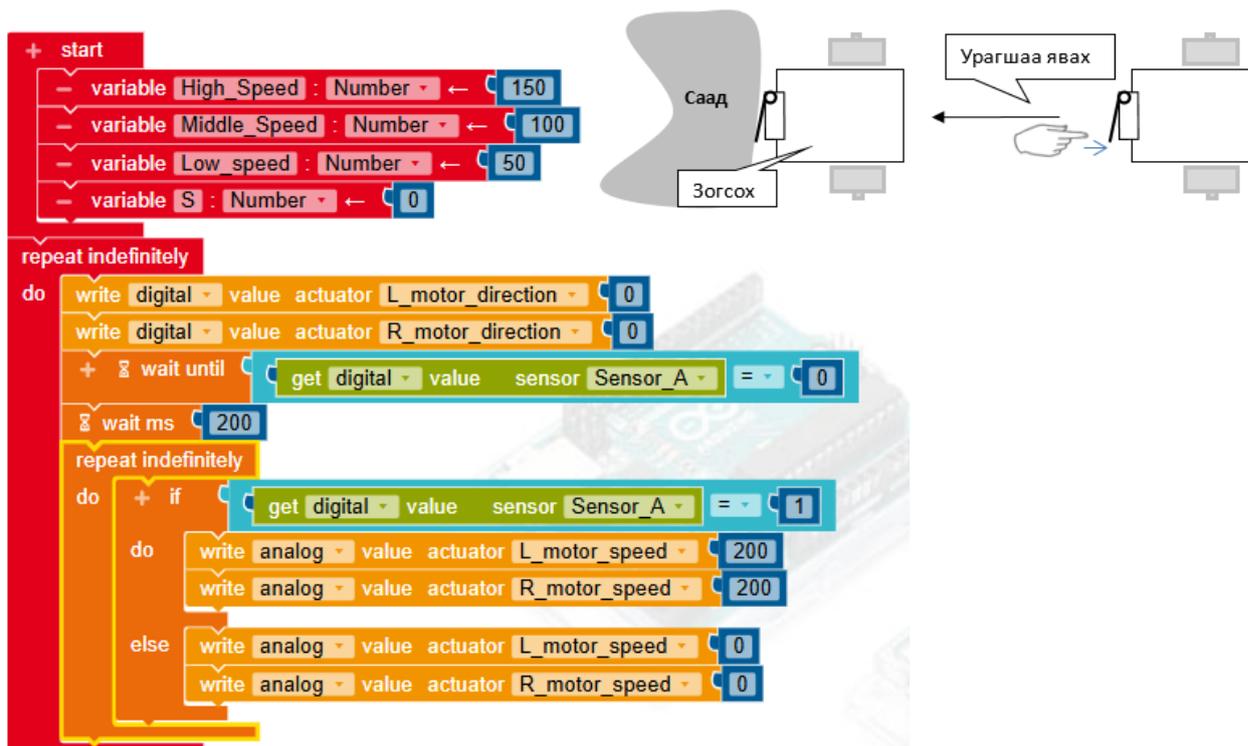
❖ Програмын код



Дасгал 4.4.2 Доорх дарааллын дагуу гүйцэтгэлийг хийж, роботын ажиллагааг туршаарай.

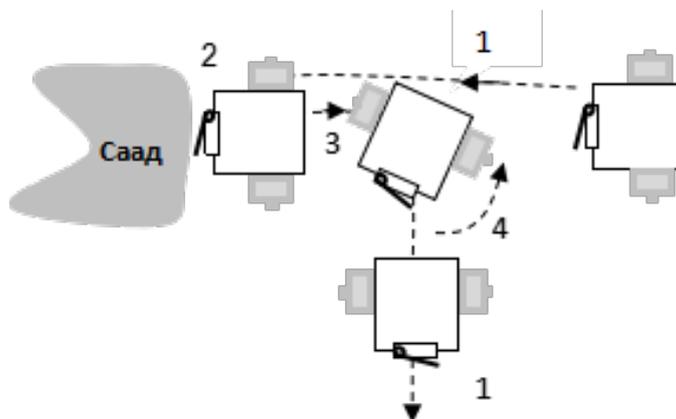
- Роботын урд талд тохиромжтой байрлалыг сонгон контактан мэдрэгчийг байрлуул.
- Доорх зургийн дагуу контактан мэдрэгчийг микрокомпьютерийн Ардуиногийн модул 2- т холбо.
- Контактан мэдрэгчийг дарах үед робот нь урагшаа явж эхлэх бөгөөд роботын өмнө объект тааралдах үед контактан мэдрэгчээр саадыг илрүүлж, роботыг зогсоох програмын кодыг бич.



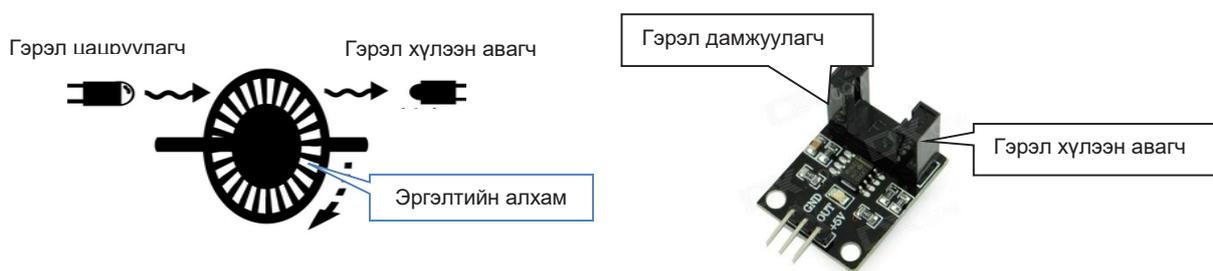


Дасгал 4.4.3 Доорх дарааллын дагуу гүйцэтгэлийг хийж, роботын ажиллагааг туршаарай.

- Роботын урд талд тохиромжтой байрлалыг сонгож, контактан мэдрэгчийг байрлуул.
- Контактан мэдрэгчийг микрокомпьютерийн модул 2-ын 5 дугаартай оролтонд холбо.
- Робот нь тэжээл залгах үед урагшаа явж эхлэх бөгөөд роботын өмнө объект тааралдах үед контактан мэдрэгчээр саадыг илрүүлж, роботыг 0.5 секунд зогсоо.
- Робот нь дахин хойшоо 0.5 секунд ухарч, дараа нь баруун ухарч эргэн чиглэлээ өөрчлөн цааш явах програмын кодыг бич.



4.5 Эргэлт мэдрэгч бүхий роботын програмчлал



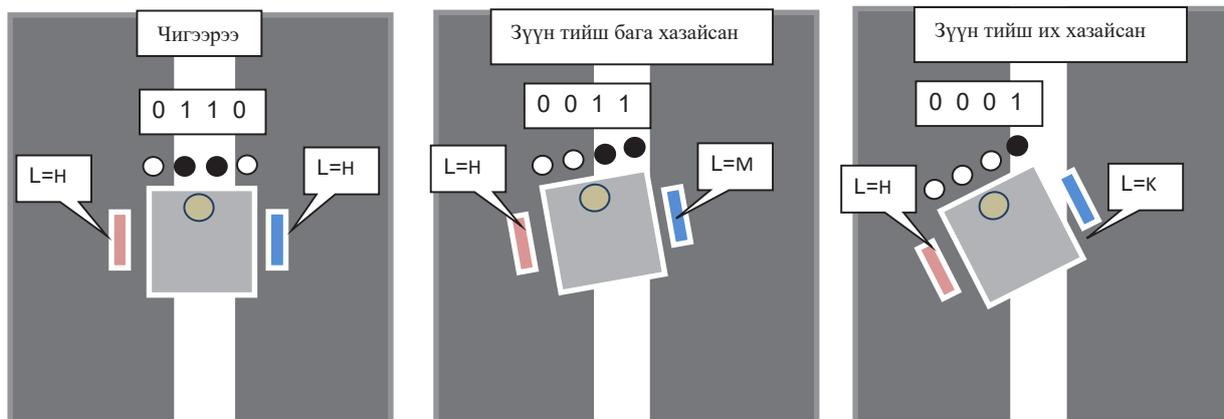
Дасгал 4.5.1 Доорх дарааллын дагуу гүйцэтгэлийг хийж, роботын ажиллагааг туршаарай.

- Моторын голд эргэлтийн мэдрэгчийг байрлуул.
- Доорх зургийн дагуу эргэлтийн мэдрэгчийг микрокомпьютерийн модул 2- т холбо.
- Робот нь тэжээл залгах үед урагшаа явж эхлэх бөгөөд эргэлт мэдрэгчийн тоололтын утгыг 10 болоход роботыг зогсоох програмын кодыг бич.

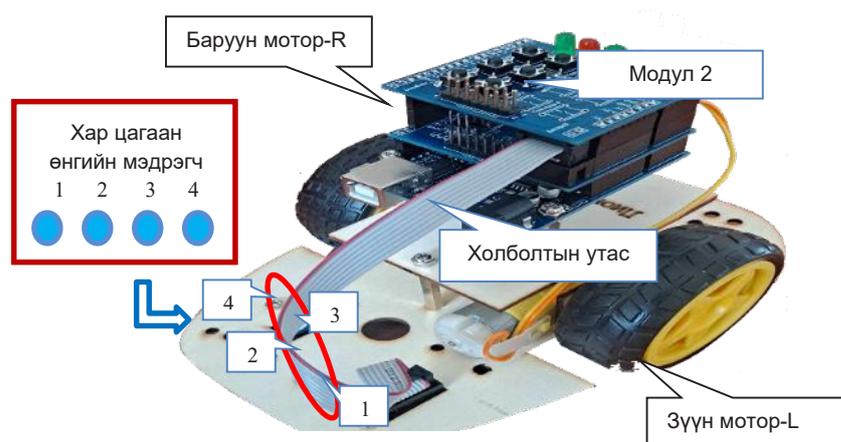


```
+ start
- variable a : Number ← 0
- variable Line_counter : Number ← 0
- variable e : Number ← 0
- variable x : Number ← 0
repeat indefinitely
do
write digital value actuator L_motor_direction 0
write analog value actuator L_motor_speed 200
write digital value actuator R_motor_direction 0
write analog value actuator R_motor_speed 200
set a to get digital value sensor Sensor_A
+ if a = 1
do
set e to 0
else
+ if e = 0
do
set x to x + 1
set e to 1
+ if x = 10
do
repeat indefinitely
do
write analog value actuator L_motor_speed 0
write analog value actuator R_motor_speed 0
```

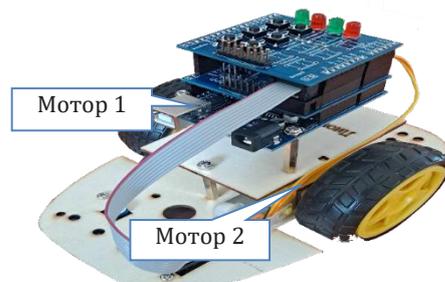
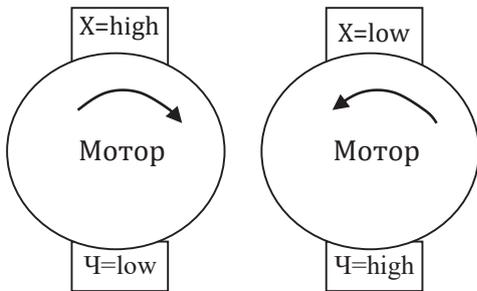
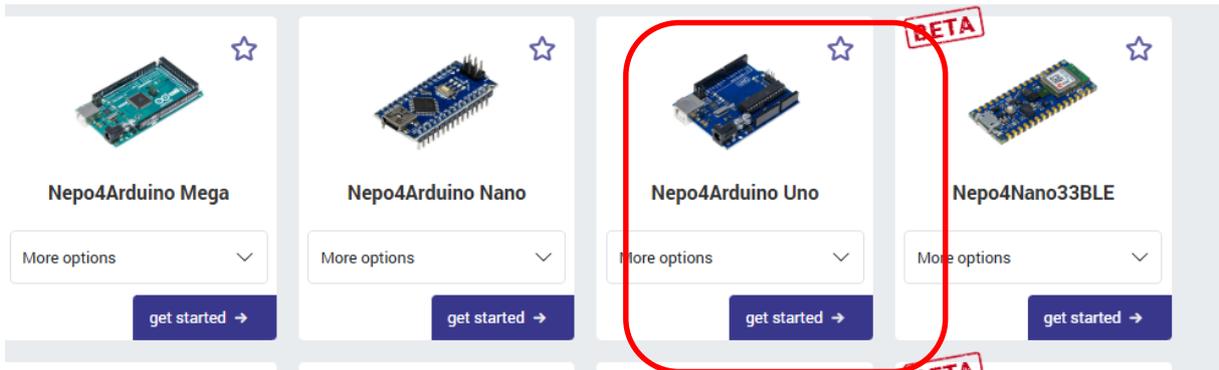
4.6. Зураас дагагч роботын програмчлал



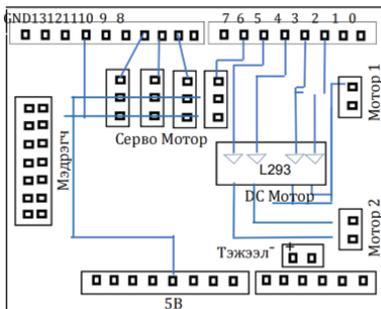
| Мэдрэгчийн дугаар | Утга | | | | 10-тын тооллын утга | Тайлбар (Обол хар1 бол цагаан) | Моторын хурдны утга (L-зүүн моторын хурд R-баруун моторын хурд H дээд хурд M дундаж хурд Q-дунд хурд) K-бага хурд) |
|--|------|---|---|----|---|-----------------------------------|--|
| | 1 | 2 | 3 | 4 | | | |
| Хувьсагч | A | B | C | D | | | |
| S-Роботын хазайлтын чиглэлийг тодорхойлох хувьсагч | 0 | 0 | 0 | 1 | 1 | Робот зүүн тийш их хазайсан | L=H :R=K |
| | 0 | 0 | 1 | 0 | 2 | Робот зүүн тийш дунд хазайсан | L=H :R=M |
| | 0 | 0 | 1 | 1 | 3 | Робот зүүн тийш бага хазайсан | L=H :R=M |
| | 0 | 1 | 1 | 0 | 6 | Робот чигээрээ явсан | L=H :R=H |
| | 1 | 1 | 0 | 0 | 12 | Робот баруун тийш бага хазайсан | L=M :R=H |
| | 0 | 1 | 0 | 0 | 4 | Робот баруун тийш дунд хазайсан | L=M :R=H |
| | 1 | 0 | 0 | 0 | 8 | Робот баруун тийш их хазайсан | L=K :R=H |
| | 0 | 0 | 0 | 0 | 0 | Робот замаасаа гарсан | L=x :H=x |
| 1 | 1 | 1 | 1 | 15 | Робот хөндлөн цагаан зураас дээгүүр явсан | C=C+1 | |



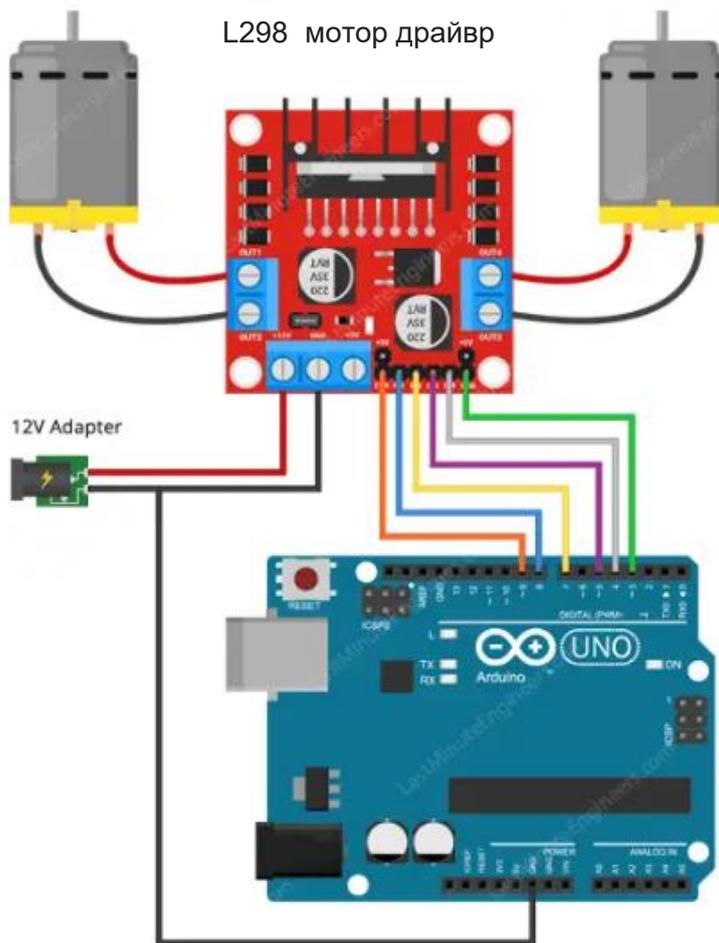
Дасгал 4.6.1 Цагаан зураасыг даган явах роботын програмын кодыг бич.



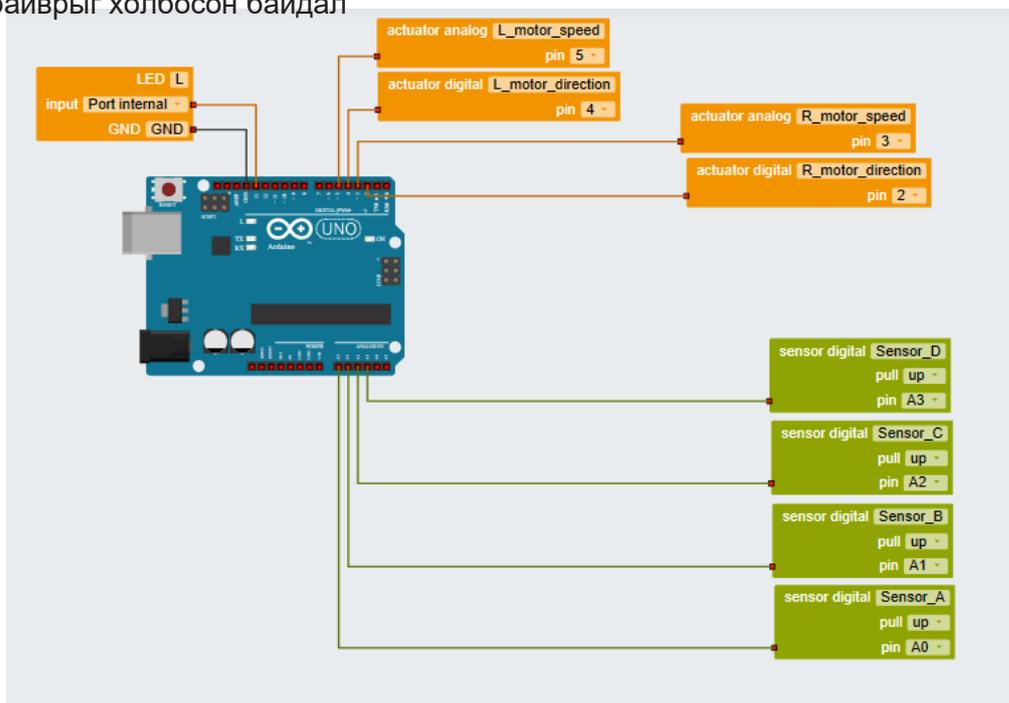
L293 Ардиуно модул 2 мотор драйвр



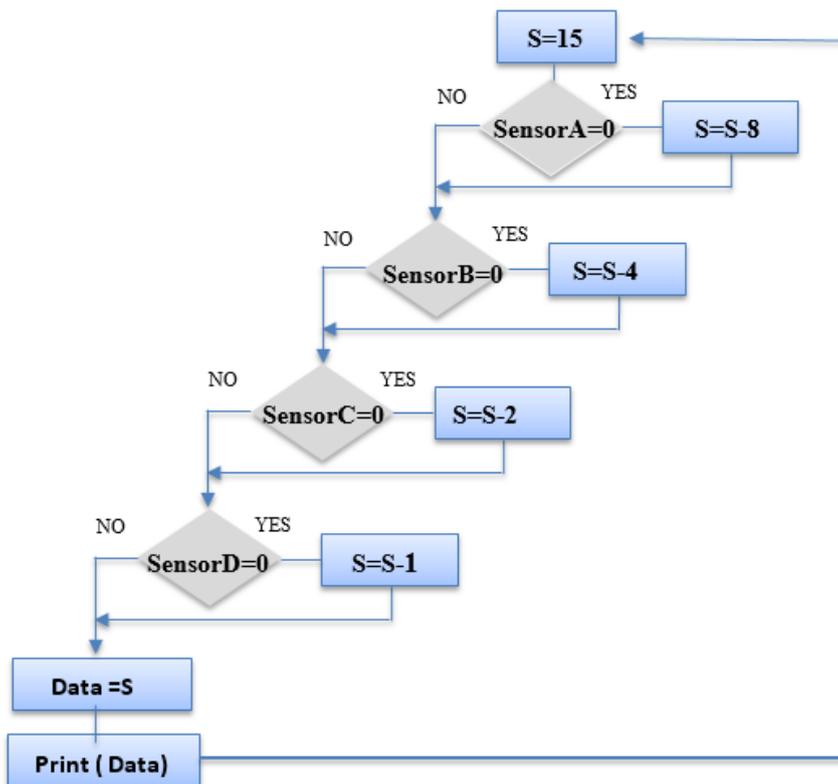
L298 мотор драйвр



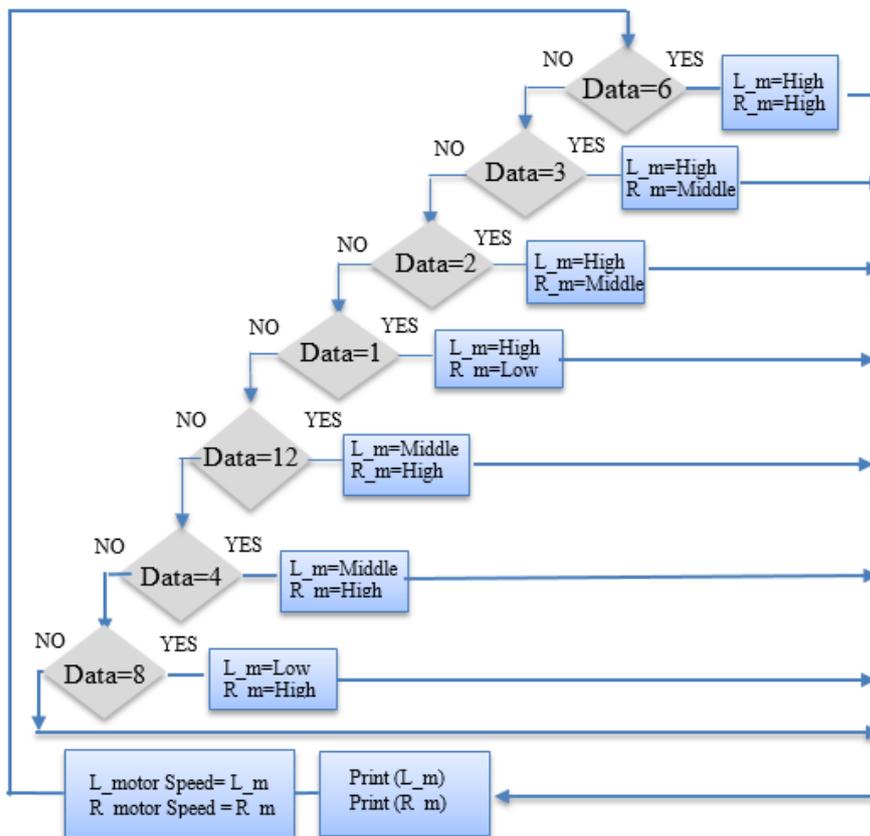
Open Roberta Sim E.V3 ийн симуляцийн орчинд L293 Ардиуно модул 2 мотор драйврыг холбосон байдал



❖ Мэдрэгчийн алгоритм



❖ Моторын удирдлагын алгоритм



```

+ start
- variable High_Speed : Number ← 150
- variable Middle_Speed : Number ← 100
- variable Low_speed : Number ← 50
- variable sensor_data : Number ← 0
- variable S : Number ← 0
- variable data_reading : Number ← 0
- variable Left_motor : Number ← 0
- variable Right_motor : Number ← 0

repeat indefinitely
do
write digital value actuator L_motor_direction 0
write digital value actuator R_motor_direction 0
set S to 15
+ if get digital value sensor Sensor_A = 1
do set S to S - 8
+ if get digital value sensor Sensor_B = 1
do set S to S - 4
+ if get digital value sensor Sensor_C = 1
do set S to S - 2
+ if get digital value sensor Sensor_D = 1
do set S to S - 1
set data_reading to S
show on Serial Monitor " Sensor_data "
show on Serial Monitor data_reading

```

```

+ if data_reading = 6
do
set Left_motor to High_Speed
set Right_motor to High_Speed
else if data_reading = 3
do
set Left_motor to High_Speed
set Right_motor to Middle_Speed
else if data_reading = 2
do
set Left_motor to High_Speed
set Right_motor to Middle_Speed
else if data_reading = 1
do
set Left_motor to High_Speed
set Right_motor to Low_speed
else if data_reading = 12
do
set Left_motor to Middle_Speed
set Right_motor to High_Speed
else if data_reading = 4
do
set Left_motor to Middle_Speed
set Right_motor to High_Speed
else if data_reading = 8
do
set Left_motor to Low_speed
set Right_motor to High_Speed
else
show on Serial Monitor " Left_motor_Speed="
show on Serial Monitor Left_motor
show on Serial Monitor " Right_motor_Speed="
show on Serial Monitor Right_motor
write analog value actuator L_motor_speed Left_motor
write analog value actuator R_motor_speed Right_motor

```



```

1. // This file is automatically generated by the Open
2.
3. #include <Arduino.h>
4.
5. #include <NEPODefs.h>
6.
7.
8. double __High_Speed;
9. double __Middle_Speed;
10. double __Low_speed;
11. double __sensor_data;
12. double __S;
13. double __data_reading;
14. double __Left_motor;
15. double __Right_motor;
16. int __output_L_motor_speed = 5;
17. int __output_L_motor_direction = 4;
18. int __output_R_motor_direction = 2;
19. int __input_Sensor_A = A0;
20. int __input_Sensor_B = A1;
21. int __input_Sensor_C = A2;
22. int __input_Sensor_D = A3;
23. int __led_L = LED_BUILTIN;
24. int __output_R_motor_speed = 3;
25. void setup()
26. {
27.   Serial.begin(9600);
28.   pinMode(__output_L_motor_speed, OUTPUT);
29.   pinMode(__output_L_motor_direction, OUTPUT);
30.   pinMode(__output_R_motor_direction, OUTPUT);
31.   pinMode(__input_Sensor_A, INPUT_PULLUP);
32.   pinMode(__input_Sensor_B, INPUT_PULLUP);
33.   pinMode(__input_Sensor_C, INPUT_PULLUP);
34.   pinMode(__input_Sensor_D, INPUT_PULLUP);
35.   pinMode(__led_L, OUTPUT);
36.   pinMode(__output_R_motor_speed, OUTPUT);
37.   __High_Speed = 150;

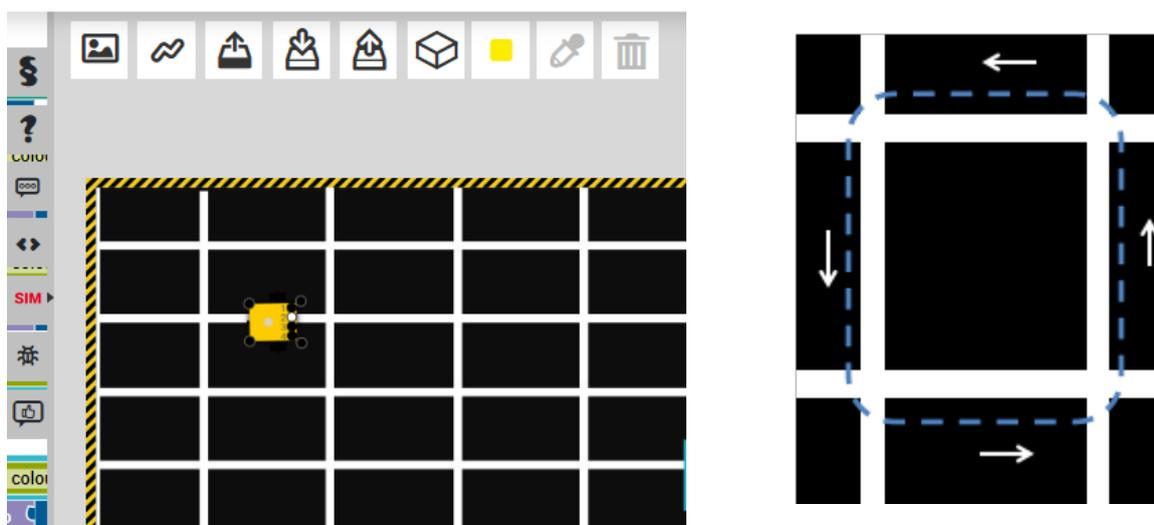
```

```

Arduino Uno
Line_robot_4.7.1.ino
1 // This file is automatically generated by
2
3 #include <Arduino.h>
4
5 double __High_Speed;
6 double __Middle_Speed;
7 double __Low_speed;
8 double __sensor_data;
9 double __S;
10 double __data_reading;
11 double __Left_motor;
12 double __Right_motor;
13 int __output_L_motor_speed = 5;
14 int __output_L_motor_direction = 4;
15 int __output_R_motor_direction = 2;
16 int __input_Sensor_A = A0;
17 int __input_Sensor_B = A1;
18 int __input_Sensor_C = A2;
19 int __input_Sensor_D = A3;
20 int __led_L = LED_BUILTIN;
21 int __output_R_motor_speed = 3;
22 void setup()
23 {

```

Дасгал 4.6.2 Open Roberta Sim E.V3 ийн симуляцийн орчинд Робот нь шулуун цагаан зураасыг даган тэгш өнцөгт үүсгэж эргэлдэн явах програмын кодыг бич.



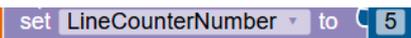
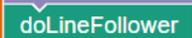
Үндсэн програмд хэрэглэгдэх туслах програмууд буюу цагаан зураасыг мэдэрч, роботыг жолоодох удирдлагын функц болон хувьсагчуудын тайлбарыг доор харууллаа.

Удирдлагыг хувьсагчууд:

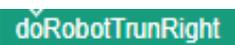
- HighSpeed –Моторын дээд хурдны утга
- MiddleSpeed–Моторын дунд хурдны утга
- LowSpeed–Моторын доод хурдны утга
- LeftMotor- Зүүн мотор
- RighMotor- Баруун мотор
- Data_Reading –Мэдрэгчүүдийн утга

Удирдлагын функцууд:

doLineFollow: Шулуун цагаан зураасын дагаж, хөндлөн цагаан зураасыг тоолж явах функц

Жишээлбэл: -   Шулуун цагаан зураасын дагаж яваад таван хөндлөн цагаан зураасыг дайрч өнгөрөн үед зогсох үйлдэл

doRobotTurnRight:– Роботыг баруун тийш эргүүлэх функц

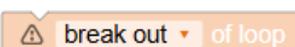
 Баруун тийш эргэн, хөндлөн цагаан зураасыг мэдрэгч дайрч өнгөрөн үед зогсох үйлдэл

doRobotTurnLeft:– Роботыг зүүн тийш эргүүлэх функц

 Зүүн тийш эргэн, хөндлөн цагаан зураасыг мэдрэгч дайрч өнгөрөн үед зогсох үйлдэл

doLineCounter:– Хөндлөн цагаан зураасыг мэдэрч тоолох функц 

doLineSensorRead:– Зураасыг мэдрэгчийн өгөгдлийг унших функц 

break out :- Үйлдлийг дуусгах команд 

```

+ start show sensor data
- variable S : Number ← 0
- variable Data_Reading : Number ← 0
- variable HighSpeed : Number ← 100
- variable MiddleSpeed : Number ← 70
- variable LowSpeed : Number ← 40
- variable LeftMotor : Number ← 0
- variable RighthMotor : Number ← 0
- variable LineCounterNumber : Number ← 0
- variable LineCounterState : Number ← 0
- variable LineCounterValue : Number ← 0
- variable SensorData : Number ← 0

repeat indefinitely
do
set LineCounterNumber to 1
doLineFollower
doRobotTrunRight

```

Үндсэн программ

```

+ doLineSensorRead
set S to 15
+ if get colour colour sensor Port 1 = 
do set S to S - 8
+ if get colour colour sensor Port 2 = 
do set S to S - 4
+ if get colour colour sensor Port 3 = 
do set S to S - 2
+ if get colour colour sensor Port 4 = 
do set S to S - 1
set Data_Reading to S

```

Зураасыг мэдрэгчийн өгөгдлийг унших туслах программ

```

+ doRobotTrunRight
turn right speed % 40
degree 90

```

Баруун тийш эргэн, хөндлөн цагаан зураасыг мэдрэгч зогсоох туслах программ

```

+ doRobotTrunLeft
turn left speed % 40
degree 90

```

Зүүн тийш эргэн, хөндлөн цагаан зураасыг мэдрэгч зогсоох туслах программ

```

+ doLineCounter
set SensorData to Data_Reading
+ if SensorData = 15
do
+ if LineCounterState = 0
do
set LineCounterValue to LineCounterValue + 1
set LineCounterState to 1
else
set LineCounterState to 0

```

Хөндлөн цагаан зураасыг мэдрэгч тоолох туслах программ

```

+ doLineFollower
  set LineCounterValue to 0
  repeat while LineCounterValue < LineCounterNumber
  do doLineSensorRead
    doLineCounter
    + - if Data_Reading = 6
    do set LeftMotor to HighSpeed
      set RighthMotor to HighSpeed
    else if Data_Reading = 3
    do set LeftMotor to HighSpeed
      set RighthMotor to MiddleSpeed
    else if Data_Reading = 2
    do set LeftMotor to HighSpeed
      set RighthMotor to LowSpeed
    else if Data_Reading = 1
    do set LeftMotor to HighSpeed
      set RighthMotor to LowSpeed
    else if Data_Reading = 12
    do set LeftMotor to MiddleSpeed
      set RighthMotor to HighSpeed
    else if Data_Reading = 4
    do set LeftMotor to LowSpeed
      set RighthMotor to HighSpeed
    else if Data_Reading = 8
    do set LeftMotor to LowSpeed
      set RighthMotor to HighSpeed
    else
    steer forwards speed % left LeftMotor
      speed % right RighthMotor
  steer forwards speed % left 0
    speed % right 0
  wait ms 1000
  
```

Дасгал 4.6.3 Робот нь зурагт өгөгдсөн замчлалын дагуу явахдаа замд тааралдсан шоог түлхэн барианы бүсэд хүргэхээр тооцоолон роботын удирдлагын програмын кодыг бич.

```

+ start
  show sensor data
  - variable S : Number ← 0
  - variable Data_Reading : Number ← 0
  - variable HighSpeed : Number ← 100
  - variable MiddleSpeed : Number ← 70
  - variable LowSpeed : Number ← 60
  - variable LeftMotor : Number ← 0
  - variable RighthMotor : Number ← 0
  - variable LineCounterNumber : Number ← 0
  - variable LineCounterState : Number ← 0
  - variable LineCounterValue : Number ← 0
  - variable SensorData : Number ← 0

repeat indefinitely
do
  set LineCounterNumber to 5
  doLineFollower
  doRobotTrunLeft
  set LineCounterNumber to 2
  doLineFollower
  doRobotTrunLeft
  set LineCounterNumber to 4
  doLineFollower
  break out of loop
  
```

The maze diagram features a grid with several obstacles. A path is indicated by dashed lines and numbered points 1 through 5. Point 1 is at the bottom right, 2 is at the middle right, 3 is at the top right, 4 is at the top left, and 5 is at the middle left. Labels in Mongolian identify obstacles: 'Шоо' (Obstacle) at the middle left, 'Бариа' (Barrier) at the bottom left, and 'Гаараа' (Hole) at the bottom right. Arrows show the robot's path starting from point 1, moving up to 2, left to 3, down to 4, and finally down to 5.

