



**BILBOKO INDUSTRIA INGENIARITZA TEKNIKOKO  
UNIBERTSITATE ESKOLA**



**INDUSTRIA ELEKTRONIKAREN ETA AUTOMATIKAREN INGENIARITZAKO  
GRADUA**

GRADU AMAIERAKO LANA

2014 / 2015

*LANAREN BESO ROBOTIKO MUGIKOR BATEN DISEINU  
PROGRAMAZIO ETA MUNTAIA*

**DISEINUA**

**IKASLEAREN DATUAK**

IZENA: ITZIAR

ABIZENAK: ALDEKOA MADARIAGA

SIN.:

DATA: 2015/06/19

**ZUZENDARIAREN DATUAK**

IZENA: M<sup>a</sup> GORETTI

ABIZENAK: SEVILLANO BERASATEGUI

SAILA: SISTEMEN INGENIARITZA ETA AUTOMATIKA

SIN.:

DATA: 2015/06/19

## AURKIBIDEA

<b>3.1. SARRERA.....</b>	<b>1</b>
<b>3.2. MUNTAKETA AURREKO PAUSOAK.....</b>	<b>1</b>
<b>3.3. PROGRAMAZIO INGURUNEA .....</b>	<b>13</b>
3.3.1 ARDUINO INGURUNEA.....	13
3.3.2. FUNTZIOAK.....	17
3.3.3. LIBURUTEGIA.....	21
<b>3.4. OINARRIA .....</b>	<b>26</b>
3.4.1. DISEINUA.....	26
3.4.2. MUNTAKETA.....	27
3.4.3. PROGRAMAZIOA.....	31
<b>3.5. BESOIA.....</b>	<b>47</b>
3.5.1. DISEINUA.....	47
3.5.2. MUNTAKETA.....	49
3.5.3. PROGRAMAZIOA.....	58
<b>3.6. OINARRIAREN ETA BESOAREN BATZEA .....</b>	<b>74</b>
3.6.1. DISEINUA.....	74
3.6.2. MUNTAKETA.....	76
3.6.3. PROGRAMAZIOA.....	80

### 3.1. SARRERA

Memoria atalean azaldu bezala proiektu honen helburua beso robotiko mugikor baten diseinu programazio eta muntaia garatzea izango da. Atal honetan zehar guzti hau egin ahal izateko emango diren pausuak deskribatuko dira.

Beso robotiko mugikorraren eginkizunak erabiltzaileak maneatuko duen aginte infragorri baten bidez kontrolatuta egongo dira. Hala ere, robotak izango duen ultrasoinu sentsore bai esker bere aurrean dauden oztopoei ihes egiteko gai izango da. Robotaren izango du eta honek bere aurrean objektu edo oztopo bat dagoela detektatzen duenean, robotak ez dio kasurik egingo erabiltzailearen aginteari eta bere mugimenduak geldituko ditu. Orduan, erabiltzaileak robotari esango dio bere aurrean dagoena oztopo edo objektua den. Horretarako, oztopoa baldin bada, botoi bati emango dio eta robota bertatik aldenduko da. Baina robotaren aurrean dagoena maneatu behar duen objektu bat baldin bada, oinarria geldirik mantenduko da eta den bitartean besoa higiaraziko da.

Baina, Robotaren muntaketarekin hasi baino lehen beharrezkoa izango da erabiliko den materiala ezagutzea. Hori dela eta, lehen esan bezala MAKEBLOCK teknologia eta Arduino programazio ingurunea erabiliko direnez hasteko MAKEBLOCK teknologiarekin proiektua garatzeko dauden piezan desberdinen deskripzioa egiten hasiko da. Eta horren ondoren, Arduinoren instalazioa eta programazio ingurunea azalduko dira.

Behin erabiliko den materiala ezagututa robota garatzen hasiko da. Robota bi zatitan eraikiko da batetik oinarria eta bestetik besoak, azkeneko pauso batean bien batuketa egingo delarik. Robotaren zatikako garapenak, atal bakoitzeko azterketa sakonagoa egiteko aukera eskainiko du. Azterketa horretan zehar, proiektu honekin bateragarriak izan daitezkeen bi beso mota aurkitu dira bata, beso tradizionala eta bestea, transmisio zuzeneko besoa. Biak baliogarriak izan daitezkeenez biak eraikitzea erabaki da, eta ondoren proiektura hobeto egokitzen dena hautatuko egingo da dela erabaki da.

Robotaren garapena oinarritik hasiko da, ondoren bi beso motak eraikiko dira. Robotaren atal bakoitzeko eraketa prozesua berdina izango da, hau da, beti hurrengo hiru pausoak garatuko dira. Lehenengo, diseinua garatu beharko da, hau da, eraikitzen ari den

robotaren zatiak izan behar dituen ezaugarriak kontuan izanik diseinurik egokiena erabaki eta definituko da pentsatuko da, diseinu hau Autodesk Inventor 3D softwarea erabiliz egingo da. Diseinua finkaturik dagoenean, muntaketa egiteko momentua da, horretarako, MAKEBLOCK teknologia erabiliko da atal horren egitura fisikoa eraikitzeke eta osagai elektronikoak kokatzeko. Bukatzeko, Arduino software-a erabiliz osagai elektronikoek programazioa garatuko da. Programa sinpleetatik hasiz, pixkanaka programazio zailduz joango da robotaren helburura iritsi arte.

Atal guztiak eraikita daudenean, robotaren bi zatien batuketa egin beharko da. Baina, hori baino lehen beso bien arteko aukeraketa egingo da. Beso bietako bat aukeratu denean hau oinarriarekin batu egingo da. Batuketa prozesu hau kontu handiarekin egin beharko da robotaren zatirik hondatu ez dadin. Baina robotaren bi zatien batuketa ez da bakarrik fisikoa izango, izan ere, oinarriak eta besoak bakoitzak bere programa garaturik izango dute, baina, robot bakarra kudeatu ahal izateko biak programa bakar bat osatu dezaten elkartu beharko dira baldin badago programa bakarra beharko da. Beraz, bi programak erabilita bakar bat eraiki beharko da eta honen funtzionamendu egokia garatu behar da, robotak bere eginkizuna bete ahal izateko.




### 3.2. MUNTAKETA AURREKO PAUSOAK

Robota muntatzen hasi orduko komeni da material guztia prest edukitzea, bai software eta bai hardwarea. Beraz, hardwarea prest edukitzeko, MAKEBLOCK-eko osagai elektronikoak eta aluminiozko pieza guztiak zenbatu behar dira. Software-ari dagokionez, Arduino software eta MAKEBLOCK-eko osagai elektronikoak kudeatzeko beharrezko izango den, MAKEBLOCK-eko liburutegiak instalatu behar dira.





Hasteko, proiektuko robota eraikitzeko dagoen aluminiozko pieza eta osagai elektronikoak zerrendatu behar dira. Horrela, robota eraikitzerakoan jakin daiteke, une oro, zenbat pieza dauden, zenbat geratzen diren eta zeintzuk dauden erabilgarri. Zerrendan osagai kopurua, osagaiaren izena, elementuaren zereginaren azaltzen duen deskribapen laburra eta bere irudia adierazten dira. Material zerrenda erabilgarriagoa izan dadin eta bertan edozein elementu errazago aurkitua ahal izateko osagaiak taldekatuta agertzen dira. Proiektua garatzeko hasierako material zerrenda hurrengo hau da:




<i>Taula 3.2.1. Plaken zerrenda.</i>			
KOP	IZENA	DESKRIBAPENA	IRUDIA
1	Arduino UNO  Arduino UNO plaka	Mikrokontroladorea da. Mikrokontroladore birprogramagarri eta hainbat sarrera/irteerei esker sentsoreak eta eragingailuak konektatu ahal diren hardware libreko plaka da.	
2	Meduino  Meduino	Arduino UNO plaka oinarritu dira Meduino plaka garatzeko. Biek ezaugarri berdinak dituzte. Irduietan ikus daitekeen bezala berdinak dira.	
3	Me Base Shield  MAKEBLOCK-eko plaka	MAKEBLOCK-eko plaka hau Arduino plakaren gainean kokatzen da eme/arra konexio bitartez. MAKEBLOCK-eko plaka honek osagai elektronikoak errazago konektatzeko eta gain korronteatatik babesteko balio du. Horrez gain, bi DC motor driver ditu.	

*Taula 3.2.2. Motorren zerrenda.*






KOP	IZENA	DESKRIBAPENA	IRUDIA
4	Motor Servo Serbomotorra	Serbomotorrak, berain ardatza 180 gradutan biratzen dute. Mota hauetako motorrak transmisio zuzeneko besoa higiarazteko erabiliko da.	
2	Motor DC 37mm 50RPM  50rpm-tako DC motorra	DC motorrak alde bietara biratzen dute, berain abiadura maximoa 50rpm-ko delarik. Motor hauek, robot tradizionalari higikortasuna emateko erabiliko dira.	
4	Motor DC 25mm 185RPM  185rpm-tako DC motorra	DC motorrak alde bietara biratzen dute, berain abiadura maximoa 185rpm-ko delarik. Motor hauek, robotaren oinarria mugitzeko erabiliko dira.	









*Taula 3.2.3. Osagai elektronikoen zerrenda.*

KOP	IZENA	DESKRIBAPENA	IRUDIA
3	Me Ultrasonic sensor  Ultrasoinua	Aurrean dagoen objektua zein distantziara aurkitzen den kalkulatzeko duen sentsorea da.	
2	Me Line Finder  Marra jarraitzailea	Sentsore hau IR led bat eta fototransistore batekin osaturik dago. Horrela, IR led-ak argia emititu eta foto transistoreak islapena jasoko du. Modu honetan, marra beltz bat atzealde zuri batean detektatu dezake edo alderantziz.	
1	Me Lego Motor Bridge  Lego Motorraren zubia	Lego motorrak eta Arduino bateragarriak izan daitezke erabiltzen da. TB6612FNG driver-a erabiltzen du eta bi lego motor kontrolatu daitezke.	
2	Me Infrared Receiver Decode  Infragorri hartzailea	Robotaren kontrola agente infragorrien bidez egiten denean, honen eta Arduinoaren arteko komunikazioa ahalbidetzen duen gailua da.	

2	Me Motor Driver DC Motorraren Driver-a	DC motorrak konektatzeko eta kontrolatzeko erabiltzen den gailua da. TB6612FNG driver-a erabiltzen du.	
2	Me Dual Servo Drive Serbomotorraren Drivea	Serbomotorrak konektatzeko eta kontrolatzeko erabiltzen den dispositiboa da. TB6612FNG driver-a erabiltzen du. Bi serbomotore independenteki kontrolatzeko aukera eskaintzen du.	
2	RJ25 modulua	Modulu honen bidez, MAKEBLOCK-eko ez diren bi osagai konektatzeko aukera eskaintzen du, beti ere, 5V-ekin elikadurarekin nahiko baldin badu.	










Taula 3.2.4. Piezen zerrenda.





KOP	IZENA	DESKRIBAPENA	IRUDIA
8	Plate 3x6 3x6ko xafla	Habeak beraien artean modu desberdinetan lotzeko erabiltzen den euskarria da.	
8	Bracket 3x3 3x3ko L xafla	Habeak beraien artean modu desberdinetan lotzeko erabiltzen den euskarria da.	
4	Bracket 3x6 3x6ko L xafla	Habeak beraien artean modu desberdinetan lotzeko erabiltzen den euskarria da.	
8	Servo Motor Bracket Serbomotor euslea	Euskarri hauek serbomotorea eusteko erabiltzen dira. Bat ardatzean ipintzen da eta bestea serboaren beste aldean. Euskarri hauek robotari mugimendua ematea ahalbidetzen dute.	
4	Bracket P3 P3 euslea	Gurpil zuriak edo ardatzak robotera konektatzeko erabiltzen den pieza.	

4	Beam 0824-096 6 zuloko habea	Habeak elkarrekin batuz robotaren egitura garatzen da. Piezak batzeko hiru aukera desberdin eskaintzen dute: zulo harilkatua, erdiko errail harilkatua edo izkina harilkatuen bidez.	
4	Beam 0824-128 8 zuloko habea	Habeak elkarrekin batuz robotaren egitura garatzen da. Piezak batzeko hiru aukera desberdin eskaintzen dute: zulo harilkatua, erdiko errail harilkatua edo izkina harilkatuen bidez.	
8	Beam 0824-144 9 zuloko habea	Habeak elkarrekin batuz robotaren egitura garatzen da. Piezak batzeko hiru aukera desberdin eskaintzen dute: zulo harilkatua, erdiko errail harilkatua edo izkina harilkatuen bidez.	
4	Beam 0824-160 10 zuloko habea	Habeak elkarrekin batuz robotaren egitura garatzen da. Piezak batzeko hiru aukera desberdin eskaintzen dute: zulo harilkatua, erdiko errail harilkatua edo izkina harilkatuen bidez.	
6	Beam 0824-192 12 zuloko habea	Habeak elkarrekin batuz robotaren egitura garatzen da. Piezak batzeko hiru aukera desberdin eskaintzen dute: zulo harilkatua, erdiko errail harilkatua edo izkina harilkatuen bidez.	
8	Beam 0808-144 9 zuloko habe fina	Habeak elkarrekin batuz robotaren egitura garatzen da. Piezak hauek, batzeko bi aukera desberdin eskaintzen dute: zulo harilkatua, edo errail harilkatua.	
8	Beam 0808- 12 zuloko habe fina	Habeak elkarrekin batuz robotaren egitura garatzen da. Piezak hauek, batzeko bi aukera desberdin eskaintzen dute: zulo harilkatua, edo errail harilkatua.	
8	Slider 256 Irristagailua	Irristagailuaren gainean habeak modu egokian kokatuz gero, alda batetik bestera mugitu daitezke.	




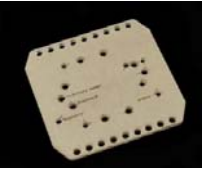





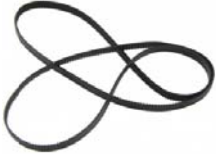
4	Gear 16T Engranaje txikia	Transmisio engranajeak motor desberdinetara, ardatzetara edo beraien artean konektatzeko erabiltzen dira. Robotari mugimendua ematen diote.	
8	Gear 48T Engranaje ertaina	Transmisio engranajeak motor desberdinetara, ardatzetara edo beraien artean konektatzeko erabiltzen dira. Robotari mugimendua ematen diote.	
8	Gear 90T Engranaje handia	Transmisio engranajeak motor desberdinetara, ardatzetara edo beraien artean konektatzeko erabiltzen dira. Robotari mugimendua ematen diote.	
1	Robot gripper Pintza	Pintza ixteko eta irekitzeko DC motor txiki bat dauka. Robotean jarri daitekeen pieza da. Objektua hartu eta maneiatzeko erabiliko da.	
8	Tire Gurpil finak	Engranajeei gurpil hauek kokatzen zaizkie robota mugitu dadin.	
4	Tire 68.5x22 Gurpil lodiak	Engranajeei gurpil hauek kokatzen zaizkie robota mugitu dadin.	
50	Track Beldar-katea	Beraien artean batzerakoan beldar-katea lortzen da, ondoren engranajetan kokatuz robotari mugimendua ematen zaio.	
12	Cutable Linkage Habe ebakigarria	Zati txikiagoko piezetan ebaki daiteke, ondoren, robotaren egitura kokatzeko.	
4	D Shaft 160mm begiratu Ardatzak luzeak	Habeak elkartzeko edo robotari mugimendua (engranajeak, ardatz konektorea...) ematen dioten piezak kokatzeko erabiltzen da.	
4	D Shaft 50mm Ardatz txikia	Habeak elkartzeko edo robotari mugimendua (engranajeak, ardatz konektorea...) ematen dioten piezak kokatzeko erabiltzen da.	

4	Threaded Shaft 4x 39mm  Ardatz harilkatua	Ardatz hauetan transmisio engranajeak jartzen dira. Eta hauek robotean egituran kokatzeko erabiltzen dira.	
4	DC Motor- 25 Bracket A  185rpm-tako DC motorraren euskarria	DC motorraren euskarri honek motorra edozein piezatan modu desberdinetan kokatzea ahalbidetzen du.	
2	DC Motor- 37 Bracket A  50rpm-tako DC motorraren euskarria	DC motorraren euskarri honek motorra edozein piezatan modu desberdinetan kokatzea ahalbidetzen du.	
2	Caster Wheel  Gurpil zuriak	Robotaren oinarrian ipinita mugimendua ematen dio robotari. Hauek biratzeko erraztasun handiagoa izango dute.	
50	Azkoinak	Azkoinen eta torloju bidez piezak beraien artean elkartzeko erabiltzen dira.	
50	Socket Cap Screw M4x8  M4x8 Allen torlojua	Torloju eta azkoinen bidez piezak beraien artean elkartzeko erabiltzen dira.	
100	Socket Cap Screw M4x14  M4x14 Allen torlojua	Torloju eta azkoinen bidez piezak beraien artean elkartzeko erabiltzen dira.	
100	Socket Cap Screw M4x22  M4x22 Allen torlojua	Torloju eta azkoinen bidez piezak beraien artean elkartzeko erabiltzen dira.	
20	Finkapen torlojuak	Piezak ardatzean zehar mugi ez daitezen jartzen den torlojua.	

15	Collar de eje Finkapen eraztuna	Piezak ardatzean zehar mugi ez daitezen jartzen den eraztuna.	
20	Separador de plastico Plastikozko zorro banatzailea	Piezak banatu eta babesteko erabiltzen den osagaia	
5	Shaft Connector 4mm Ardatz konektatzailea	Ardatzak eta engranajeak batzeko erabiltzen dira. Engranajera konektatzeko bi M4x14 allen torlojua erabiltzen dira eta ardatzean, ordea, finkapen torlojua. Horrela, engranajeek ardatzarekin batera biratuko dute.	
20	Flange Bearing 4x8x3mm Brida kojinetea	Engranaje eta ardatzaren artean kokatzen den pieza da.	

Taula 3.2.5. Bestelako elementuen zerrenda.

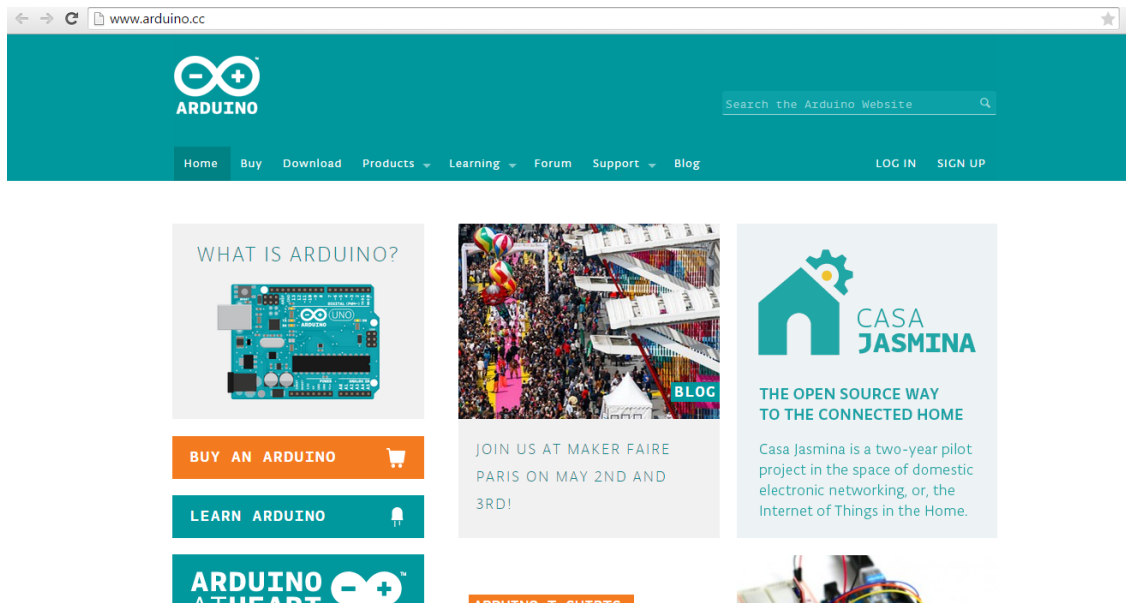
KOP	IZENA	DESKRIBAPENA	IRUDIA
30	RJ cable 20mm RJ kableak	MAKEBLOCK-eko plaka eta osagai elektronikoen artean konexioak egiteko erabiltzen diren kableak.	
1	USB cable USB Kablea	Programa plakara kargatu ahal izateko, Arduino plaka ordenagailuarekin konektatzeko erabiltzen duen kablea.	
1	Battery Holder Pila kaxa	Robota martxan ipintzeko eta ibiltzeko beharrezko potentzia (9V) ematen duen bateria.	
1	Plaka euskarria	Arduino UNO eta MAKEBLOCK-eko plaka kokatzeko euskarria, modu honetan robotaren edozein tokitan jarri daiteke.	

1	Mando infragorria	Robotari uhin infragorri bidez aginduak bidaltzeko erabiltzen den gailua.	
4	Timing Belt Transmisio uhala txikia	Engranajeen arteko transmisio egiteko erabiltzen den uhala, hauek robotari mugimendua emateko.	
2	Timing Belt Transmisio uhala ertaina	Engranajeen arteko transmisio egiteko erabiltzen den uhala, hauek robotari mugimendua emateko.	
2	Timing Belt Transmisio uhala handia	Engranajeen arteko transmisio egiteko erabiltzen den uhala, hauek robotari mugimendua emateko.	

Ondoren, proiektuan erabiliko den software-a instalatuko da. Oso erabilgarria da ezer egiten hasi baino lehen behar izango diren erreminta guztiak instalaturik edukitzea. Horrela, robota eraikita dagoenean ez da prozesua eten behar softwarearen instalazioa egiteko eta aurrera jarraitu ahalko da inongo eragozpenik gabe. Hemendik aurrera eta atal honetan zehar, proiektua garatzeko erabiliko den softwarea eta beharrezko diren liburutegien instalazioak nola egin behar diren azalduko da.

Proiektua garatzeko erabiliko den softwarea Arduino da. Arduino software librea denez ez da inolako arazorik egongo programa deskargatu eta instalatzeko. Arduinoren web orri ofizialera joan behar da eta bertatik programa deskargatu. Behin Arduino instalatuta dagoenean, MAKEBLOCK-eko materiala erabiliko denez, liburutegi berezi bat jaitsi behar da MAKEBLOCK-eko web orri ofizialetik. Liburutegi horretan, osagai elektronikoko bakoitzarentzako beharrezkoak diren komandoak aurkitzen dira. Hurrengo atalean, pausoz pauso Arduino eta MAKEBLOCK-eko liburutegien instalazioa nola egin behar den deskribatuko da.

1.- Arduino web orri ofizialean sartu (<http://www.arduino.cc/>). Bertan, Download estekan klikatu eta ondorengo orrira bideratzen du:



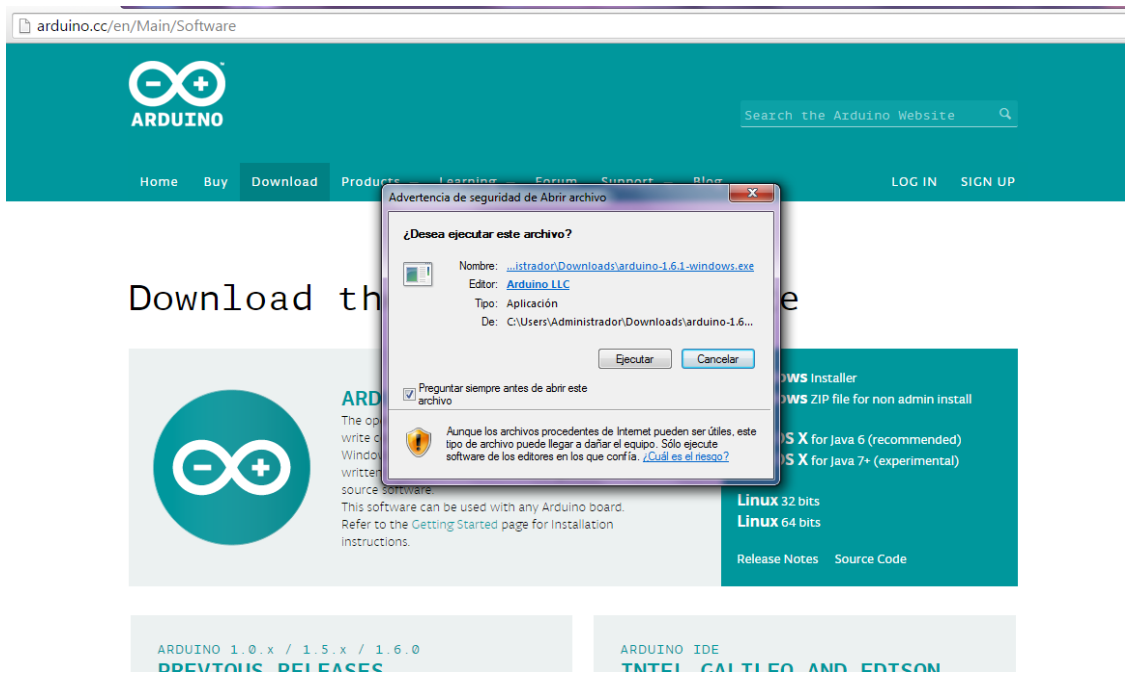
*Irudia 3.2.1. Arduinoko web orria*

2.- Eskumako aldean, Arduinoaren softwarea deskargatzeko aukera desberdinak aurkitzen dira ordenagailuko sistema eragilearen arabera, beharrezkoa dena aukeratu behar da, kasu honetan Windows. Gainera klikatzerakoan deskargatzen hasiko dela ikus daiteke.



*Irudia 3.2.2. Arduinoko web orria download*

3.- Behin deskargatu denean, automatikoki irekitzen da exekutatzeko aukera eskainiz. Exekuzio botoia sakatu eta Arduino software-aren instalazioa exekutatzen hasiko da.



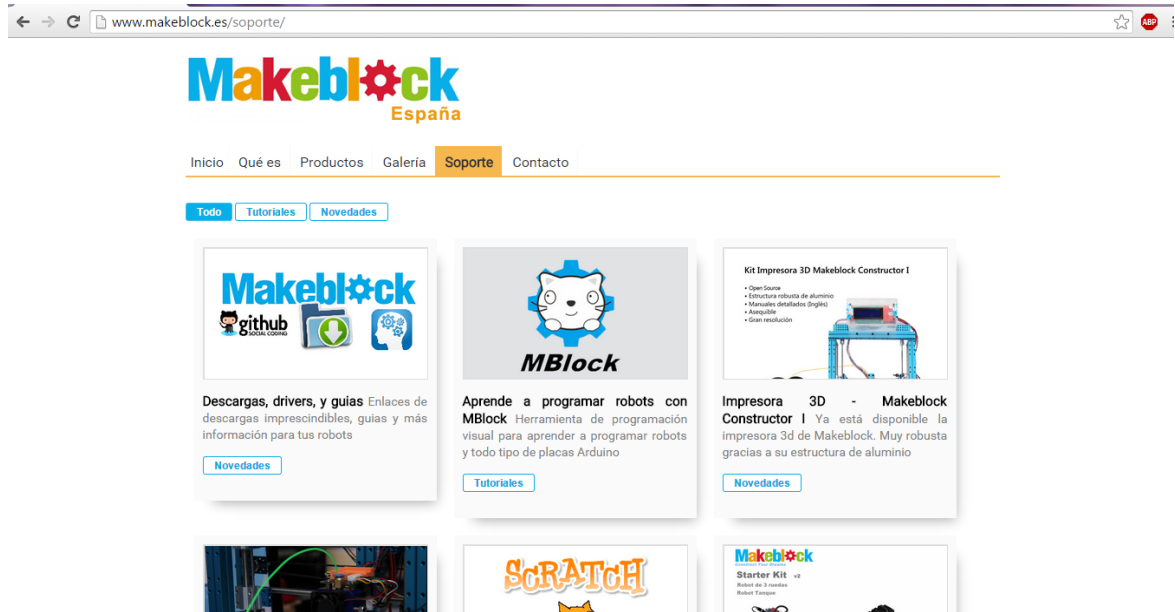
*Irudia 3.2.3. Arduino web orriaren software-a deskargatzen*

4.- Arduino instalatuta dagoenean, MAKEBLOCK-eko web orri ofizialera joan behar da. (<http://www.makeblock.es/>)



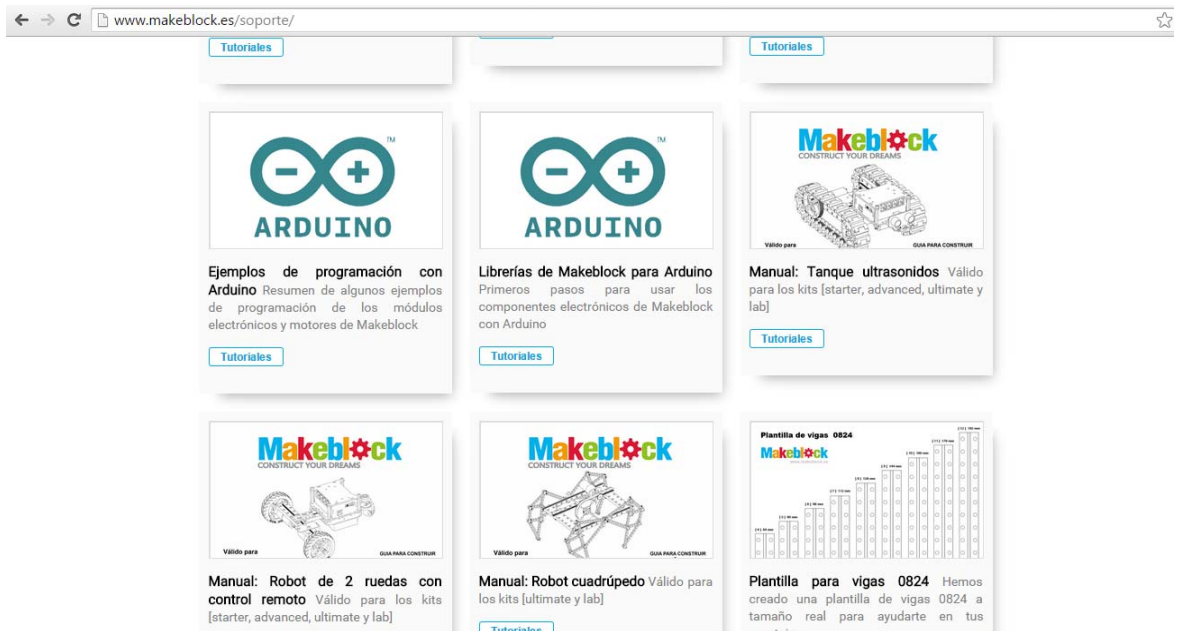
*Irudia 3.2.4. MAKEBLOCK-eko web orria*

5.- Goiko aldean “Soporte” jartzen duen estekan klikatuko da.



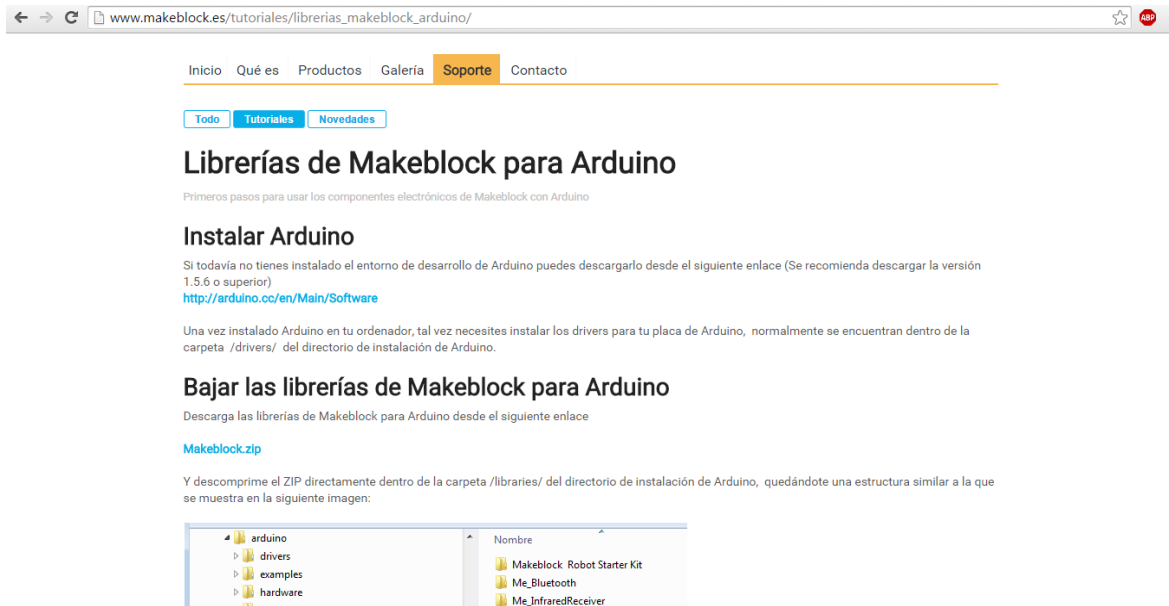
Irudia 3.2.5. MAKEBLOCK-eko web orrian “Soporte” esteka barruan

6.- Irteten den orrian beherantz joan “librerías de makeblock para Arduino” aurkitu arte.



Irudia 3.2.6. MAKEBLOCK-eko liburutegiak deskargatzeko instrukzio eta link-ak

7.- Ondoren, gainean klikatu. Hori egiterakoan hurrengo orrialdera zuzentzen du.



Inicio Qué es Productos Galería **Soporte** Contacto

Todo Tutoriales Novedades

## Librerías de Makeblock para Arduino

Primeros pasos para usar los componentes electrónicos de Makeblock con Arduino

### Instalar Arduino

Si todavía no tienes instalado el entorno de desarrollo de Arduino puedes descargarlo desde el siguiente enlace (Se recomienda descargar la versión 1.5.6 o superior)  
<http://arduino.cc/en/Main/Software>

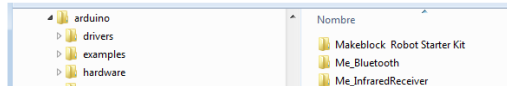
Una vez instalado Arduino en tu ordenador, tal vez necesites instalar los drivers para tu placa de Arduino, normalmente se encuentran dentro de la carpeta /drivers/ del directorio de instalación de Arduino.

### Bajar las librerías de Makeblock para Arduino

Descarga las librerías de Makeblock para Arduino desde el siguiente enlace

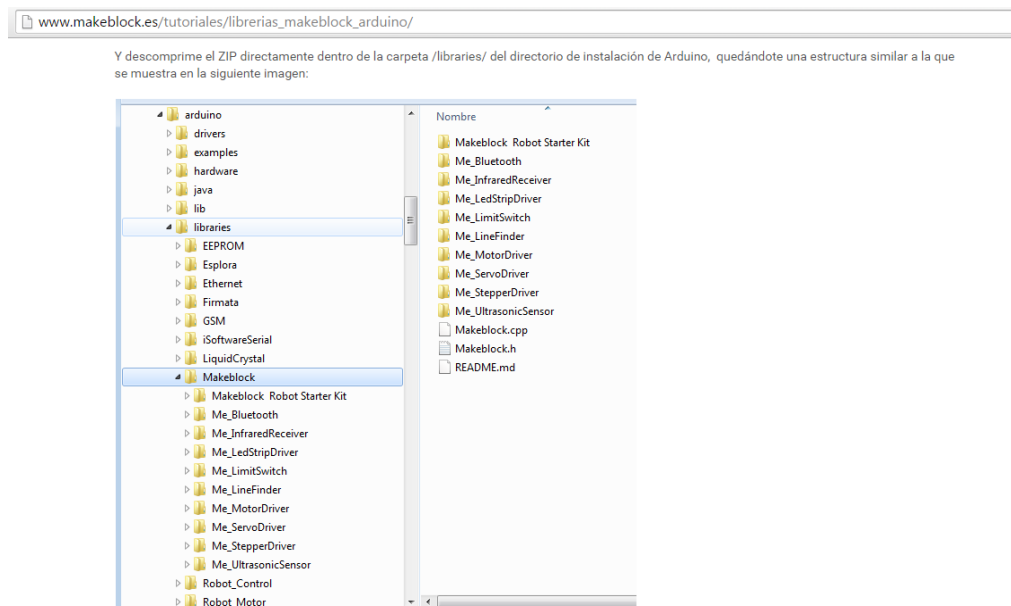
[Makeblock.zip](#)

Y descomprime el ZIP directamente dentro de la carpeta /libraries/ del directorio de instalación de Arduino, quedándote una estructura similar a la que se muestra en la siguiente imagen:



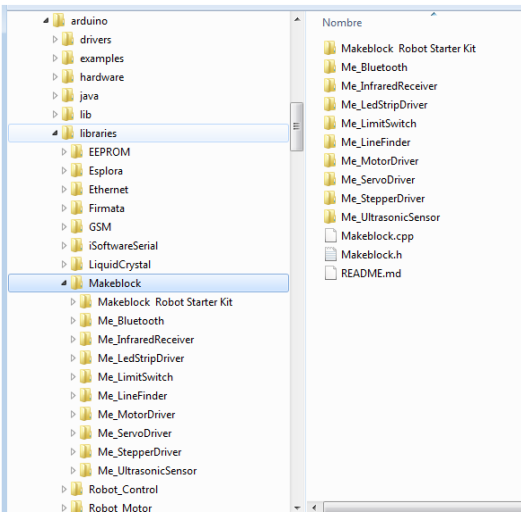
*Irudia 3.2.7. MAKEBLOCK-eko web orrian, liburutegien bila*

8.- Orrialde horretan MAKEBLOCK liburutegia deskargatzeko eta instalatzeko pauso guztiak azaltzen dira. Lehenengo Makeblock.zip fitxategia deskargatu behar da eta Arduinoko “libraries” karpeta barruan gorde. Ondorengo irudian MAKEBLOCK liburutegiaren karpeta non kokatzen den ikus daiteke.



[www.makeblock.es/tutoriales/librerias\\_makeblock\\_arduino/](http://www.makeblock.es/tutoriales/librerias_makeblock_arduino/)

Y descomprime el ZIP directamente dentro de la carpeta /libraries/ del directorio de instalación de Arduino, quedándote una estructura similar a la que se muestra en la siguiente imagen:



*Irudia 3.2.8. MAKEBLOCK-eko liburutegien kokapena*



### 3.3. PROGRAMAZIO INGURUNEA

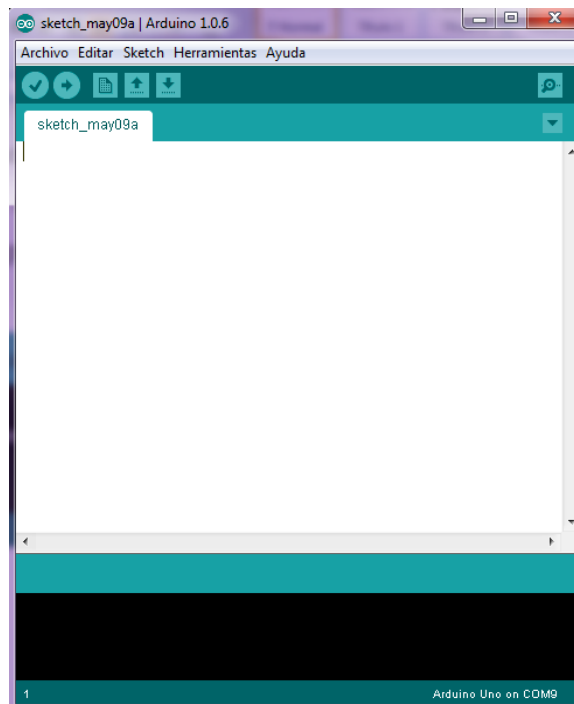
Arduino software libreko programa da. Hau da, Arduino software-a edozein erabiltzailearen esku dago bere web orri ofizialea. Doako eta mutiplataformadun Softwarea da.

Arduinorekin lan egitean askotan Arduino IDE (Interated Development Enviroment) erabiltzen da, Arduino Garapen Integratuko Ingurunea. IDE programatzaileek programa garatzeko erabiltzen duten software-eko erreminta multzoari deritzo. Programa idatzi, editatu, konpilatu eta errorerik egotekotan hauek zuzentzea ahalbidetzen duen ingurunea da, behin zuzenduta dagoenean Arduinoko plakako mikrokontrolagailura kargatzeko erabiltzen da Arduino IDE-a. Guzti honez gain, aipatu beharra dago Arduino ingurunean, exekutatzeko diseinaturiko programa sketch izenez ere ezagutzen direla.

Proiektu honetako robota programatzeko erabiliko den Arduino IDE-ari buruzko oinarritzko ezagutza batzuk beharko dira. Hori dela eta, atal honetan robotaren programazioa garatzeko beharrezkoak diren programazio erremintak azalduko dira. Hauek hiru ataletan banatu dira: Arduino ingurunea, funtzioak eta MAKEBLOCK-eko liburutegiak. Arduino software-aren inguruneari dagokionez, garrantzitsua izango da software ingurunearen ezaugarriak eta funtzionamendua ezagutzea proiektuan erabiliko diren programak garatzeko. Ondoren, funtzioak azalduko dira. Funtzioak hainbat programazio lengoaietan erabiltzen diren elementuak dira eta atal honen helburua programak garatzerakoan hauek modu egokian erabiltzeko aholkuak eta deskribapenak ematea da. Azkenik, MAKEBLOCK liburutegi atalean, liburutegi honen bidez, MAKEBLOCK-eko osagai elektronikoak programatzeko erabiliko diren komandoak deskribatuko dira.

#### 3.3.1 ARDUINO INGURUNEA

Atal honetan zehar, Arduino software librearen programazio ingurunea deskribatuko da. Beraz, beharrezkoa izango da Arduino software-a irekitzean agertzen den pantaila nagusiko elementuak azaltzea, beraien erabilera ezin hobea lortzeko helburuarekin.



*Irudia 3.3.1. Arduinoren pantaila nagusia*

3.3.1 Irudian Arduino softwarea irekitzean agertzen den pantaila ikusi daiteke. Pantaila hau bost ataletan banatuta dago, goitik hasita eta behera iritsi arte menu-barra, botoi-barra, kode-editorea, barra eta mezuen kontsola aurkitzen dira.

Menu-barra: Menu barraren barruan bost botoi daude: “Archivo”, “Editar”, “Sketch”, “Herramientas” eta “Ayuda”. Banan-banan bakoitza aztertzen eta beraien barruko aukera desberdinak ikusiko dira.

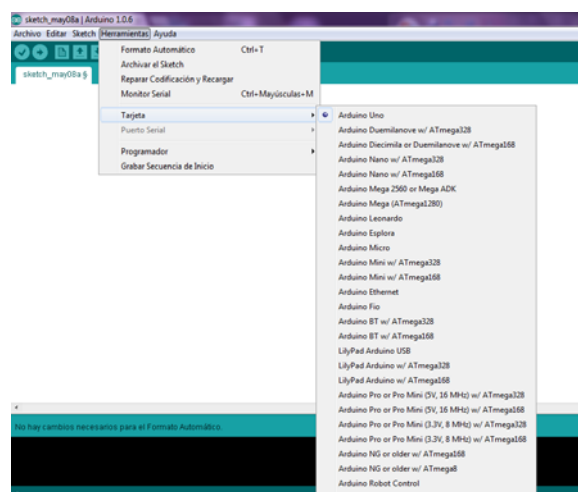
“Archivo” fitxategiarekin erlazionaturiko aukerak eskaintzen ditu, esate baterako, fitxategi berri bat ireki, gorde, programa inprimatu, programatik irten, programa kargatu, itxi, adibideak ...

“Editar” sakatzerakoan programaren itxura aldatzeko erabilgarriak izango diren aukerak eskaintzen dira. Berregin, desegin, kopiatu, ebaki, itsatsi, guztia aukeratu, bilatu, marjina handitu, marjina txikitu, eta iruzkinak sartu/kendu. Azkeneko hau, behar bada interesgarriena da horrela programa errazago ulertzeko azalpen txikiak ipintzeko aukera edukiko da.

“Sketch” programa modu egokian garatu ahal izateko aukerak eskaintzen ditu. Barruan egiaztatu/konpilatu aukerak daude, hauek botoi-barran ere aurkitzen direlarik, artxiboa gehitu, liburutegia inportatu eta sketch-a gordeta dagoen karpeta erakusteko aukera aurkitzen da.

“Ayuda”-ren barruan laguntza mota desberdinak ageri dira, hauek Arduinoko web orrira zuzentzen duten estekak dira.

“Herramientas” barruan programa modu egokian kargatu ahal izateko elementuak daude. Bertan formatu automatikoa, sketch artxibatu, sketch-a .zip dokumentu batean gordetzen du. Horrela kodea konpondu eta berriz kargatu daiteke. Beharbada, talde honetatik garrantzitsuenak, “Tarjeta” ipintzen duen botoia izango da. Sagua honen gainean ipiniz gero zerrenda luze bat irekiko da, aurki daitezkeen Arduino txartel mota desberdinekin. Edozein programa garatu baino lehen, zerrendatik erabiliko den txartela aukeratu behar da. Proiektu hau garatzeko erabiliko den txartela Arduino UNO denez honen gainean emango zaio aukeratzeko.



*Irudia 3.3.2. Arduinon txartelen aukera desberdinak*

**Botoi-barra:** botoi-barran zeregin desberdinak dituzten sei botoi desberdin aurkitzen dira.



Lehenengoa *egiaztatu* botoia da. Honi ematean, kodean akatsik ez dagoela egiaztatzen da. Kodea zuzena bada, konpilatu egingo du, baina kodea

zuzena ez bada abisatu egingo du. Kode bat egin ostean edo aldaketak egin ostean beti botoi honi ematea komeni da kodearen akatsak kentzeko.



Hurrengo botoia *kargatu* da. Botoi honen funtzioa, konpilaturiko sketch-a Arduino plakako mikrokontrolagailuko memorian gordetzea da.



*Berria* botoiari ematean, hutsik dagoen sketch berria bat irekitzen da.



*Ireki* botoiari ematean, irekitzeko prest dauden programak agertuko dira. Programa hauek adibideak edo bakoitzak eginiko izan daitezke.



*Gorde* botoiarekin sketch-an garaturiko kodea Arduino fitxategi batean, hau da .ino fitxategia batean gordeko da. Fitxategi hau izen berdina duen karpeta baten barruan gordeko da. Edozein tokitan gorde daiteke, baina, Arduinok “sketchbook” karpetan gordetzeko aukera eskaintzen du.



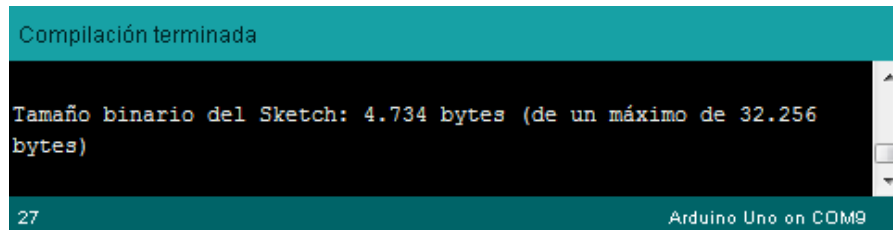
Azkeneko botoia eskuma aldean ageri da. Botoi honekin serie monitorea irekitzen da.

Kode editorea, pantaila nagusi erdian dagoen atal zuria da. Pantaila horretan programa edo sketch idazteko erabiltzen da.

Mezuen kontsola behe aldean aurkitzen den karratu beltza izango da. Mezuen kontsolan, konpilatzerakoan sketch-eko kodekoan errorerik dagoen edo ondo konpilatu eta kargatu den adieraziko da. Errorerik badago, letra laranja errorea zein den eta zein lerrotan aurkitzen den adieraziko du. Ondo konpilatu bada, mikrokontroladorearen Flash memorian, sketch-ak beteko duen tamaina agertzen da. Informazio hau oso erabilgarria da, plakak jasan dezakeen tamainatik hurbil balego, kodea tamainaz txikitzeko moldatu ahal izango litzateke, limitetik ez pasatzeko. Hasieran esan bezala, proiektua garatzeko 32KB-ko memoria duen Arduino UNO plaka erabiliko da eta. Beraz, datu hau kontutan izan beharko da, konpilatzerakoan.

Barra mezuen kontsolaren gainean eta azpian aurkitzen diren bi karratu berdeak/urdinak dira. Goiko barran, prozesuen egoera denbora errealean adieraziko da, hau

da, sketch-a konpilatzen, kargatzen... dagoen. Beheko barran, aldiz, eskumako aldean momentu horretan kurtsoa kodeko zein lerrotan dagoen adierazten da eta ezkerreko aldean erabiltzen ari den Arduino plaka zein den eta zein portutara konektatu dagoen.



*Irudia 3.3.3. Arduinoko menu kontsola eta barra*

### 3.3.2. FUNTZIOAK

Arduino programazio ingurunea aukera ugari eskaintzen ditu. Atal honetan bakarrik proiektuan zehar programatzeko erabiliko diren funtzioak azalduko dira gehiegi ez luzatzeko. Ohiko funtzioez gain, Arduinon sketch bat garatzean honen egituran ipini behar diren *void setup()* eta *void loop()* ere azalduko dira. Izan ere, hauek oso garrantzitsuak izango da programa modu egokian eraikitzeko.

#### EGITURA

Arduino programazio lengoaiaren oinarritzko egitura, bi zatitan antolatzen da. Biak beharrezkoak dira programak funtziona dezan eta ondorengo moduan antolatzen dira.

```
void setup()
{
}
void loop()
{
}
```

*Irudia 3.3.4. Arduinoko programa baten zatiak*

*Void setup()* behin bakarrik exekutatzen da, Arduino plaka pizterakoan. Hasierako aurrekonfigurazioak zehazteko erabiltzen da, hau da, sarrera irteerak definitzeko (pinMode) eta serieko komunikazioa hasteko.

*Void loop()* void setup-en ostean, behin eta berriz exekutatzen da plaka itzali arte. Hau da, void loop-eko instrukzioak exekutatzen dira lehenengotik hasi eta azkenekora arte, azkeneko instrukziora iristean berriz ere hasierakora bueltatuko da. Hemen idatzitako instrukzio multzoek programa osatzen dute.

### *IRUZKIN edo AZALPENA*

Edozein programazio lengoian bezala, programan zehar ipinitako azalpenek programa ulertzen laguntzen dute bai sortzaileari baita beste edozein erabiltzaileri. Erabileraren arabera iruzkinak gehitzeko bi ikur erabili daitezke:

```
// erabiltzen bada, azalpena lerro batekoa edo gutxiagokoa izango da
```

```
/* hasiera eta */ bukaeran ipiniz gero, bi hauen artean nahi den beste azalpen ipini daiteke. Ikur hauen bidez, lerro bat baino gehiago behar dituzten azalpenak gehitu daitezke.
```

### *ALDAGAIA*

Programan aldagaiak izendatu daitezke, aldagaiak balio bat gordetzeko erabiltzen diren elementuak dira, hala ere, sketch-ko edozein unetan balioa aldatu ahal zaio. Aldagai honen balioa modu desberdinetan ezarri daiteke: balioa zuzenean ematea edo sentsore batetik informazioa jasoz edo kalkulu bateko emaitza hartuz...

Aldagai bat sortzerakoan zein motatakoa den eta bere izena adierazi behar dira. Aldagai mota gordeko den datuaren arabera hautatuko da, hau da, zenbaki osoa, hamartarra, karaktere katea... Aldagaia izendatzeaz gain balio ezarri ahal zaio edo bestela programan zehar aurrerago ipiniko zaio.

```
Aldagai_mota aldagai_izena = balioa;
```

*Irudia 3.3.5. Aldagaien izendatzea Arduinon*

### DATU (ALDAGAI) MOTAK

Aldagai mota asko egon arren proiektu honen programak garatzeko nahikoa izango da byte eta zenbaki osoa (int) erabiltzea. Beraz, bi datu mota hauen ezaugarriak azalduko dira ondorengo lerroetan.

Byte: 0 eta 255 arteko balio oso positiboak gordetzeko erabili daiteke. Datu mota hau gordetzeko memorian 8 bit behar dira.

Int:  $-32768 (-2^{15})$  eta  $32767 (2^{15}-1)$ -ren arteko zenbaki osoak gordetzeko erabiltzen da. Datu mota honetako balioak memorian gordetzeko 2 byte (16bit) erabili behar dira.

### DENBORA

Programa garatzerakoan beharrezkoa izango den beste funtzio bat denbora izango da. Denborarentzako aukera desberdinak daude, baina, proiektu hau garatzeko delay() erabiliko da. Delay()-aren parentesien artean ipinitako denbora milisegundotakoa da. Beraz, agindu hau exekutatzen denean, programa denbora horretan zehar gelditu egingo da, itxaroten.

```
delay(1000) //segundo batez itxarongo du
```

*Irudia 3.3.6. Denbora funtzioaren erabilera Arduinon*

### BLOKEAK

Robotaren programazioa garatzeko beharrezkoak izango dira ondorengo bloke funtzio hauek erabiltzea.

if/else

*If* blokea, baldintza baten arabera zein kode zatia (aginduak) exekutatu behar diren aukeratzeko erabiltzen da. Baldintza betetzen bada, baldintzaren barruko instrukzioak exekutatu dira. Baldintza betetzen ez bada, posible da ezer ez gertatzea edo *else* blokea exekutatzea. *Else* blokea *if*-en ostean ipiniko da beti eta honen barruan beste instrukzio batzuk egongo dira.

```
if(baldintza)
{ // instrukzioak
}
else
{//instrukzioak
}
```

**Irudia 3.3.7.** *If/else* egitura Arduinon

Aukera bat baino gehiago badaude, *if*-aren ostean nahi beste *else if* ipini daitezke. *Else if*-ak erabiltzean ez da beharrezkoa sketch-aren bukaeran *else* ipintzea.

```
if(baldintza)
{ // instrukzioak
}
else if(baldintza)
{//instrukzioak
}
else if(baldintza)
{//instrukzioak
}
else
{//instrukzioak
}
```

**Irudia 3.3.8.** *If/else if* egitura Arduinon



### switch

Hainbat baldintza segidan ipini behar direnean aukerarik egokiena da. *Switch* blokea, *if/else* antzekoa da, baina, guztia konpaktuago eta txukunago antolatzea posible egiten du. *Switch* blokearen barruan *case* eta aukerakoa den *default* aurkitzen dira. *Switch*-eko lehenengo lerroa heltzean, aldagaiaren balio edo parentesien arteko adierazpena egiaztatzen da. Balioa *case*-eko lehenengoren berdina bada, *case* -aren barruko instrukzioak exekutatzen dira, eta *switch* bukatutzat ematen da. Lehenengoa *case* -aren emaitzarekin bat ez badator, hurrengoa begiratzera pasako da, modu horretan jarraituko du bukaerara arte. Amaieran, *default* jartzeko aukera dago, baina, ez da derrigorrezkoa. *Default*-en sententziak exekutatuko dira aurreko *case* baldintzetako bat bera bete ez bada.

*Break* instrukzioa begizta baten barruan ipini behar da, helburua bere aurreko instrukzioak exekutatzean *switch* buklea berehala bukatzea da. Hau da, begiztaren ostean programa exekutatzen jarraitzea derrigortuko du.

```
switch(baldintza)
{
  case :
    //instrukzioak
    break;
  case :
    //instrukzioak
    break;
  case :
    //instrukzioak
    break;
}
```

*Irudia 3.3.9. Switch egitura Arduinon*

### 3.3.3. LIBURUTEGIA

MAKEBLOCK-etik deskargatutako liburutegian osagai elektronikoak programatzeko komandoa bereziak daude. Proiektuan zehar garatuko den programa hobeto ulertzeko eta

hauen erabilera egokia bermatzeko helburuarekin, erabiliko diren osagai elektroniko bakoitzak dituen komandoa azalduko dira.

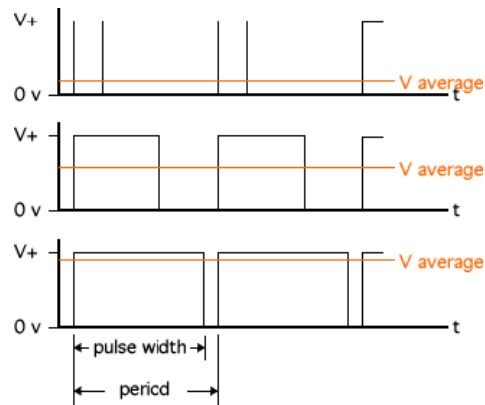
### DC MOTORRA ETA DRIVERRA

*MeDCMotor motor\_izena(PORT\_Num)*: komando honekin motorra izendatuko da eta zein portutan konektatutik dagoen adieraziko da. Motorren konexioak parentesien artean PORT\_1, PORT\_2 bakarrik izan daitezke. Motorra, driverrera konektaturik egon beharrean, motorra plakako driver-retara zuzenean konektatzen bada M1 edo M2 izango litzateke.

*Motor\_izena.stop()*: motorra gelditu egingo da.

*Motor\_izena.run(int motorSpeed)*: motorrek biratze abiadura parentesi artean zehaztuko da. Abiadura positiboa bada alde batera biratuko du eta negatiboa bada bestera. Motor hauen abiadura ezartzeko PWM seinaleak erabiltzen dira, hori dela eta, osagai elektronikoaren komando ez izan arren beharrezkoa da hau osatzeko, beraz, hurrengo lerroetan PWM-aren nondik norakoa azalduko da.

PWM-a seinale analogikoa dela simulatzen duen seinale karratu digitala da. Seinale analogiko simulatuaren balioa, seinale digitalaren pultsuaren iraupenarena izango da, hau da, PWM-aren balio altuaren arabera izango da. PWM baten irteerak 8 bit-eko bereizmena du, beraz, seinale digitalaren balio maximoa  $2^8=255$ -ekoa izango da. Seinale analogikoan, aldiz, plaka eskaintzen duen balio maximoa 5V-etakoa da eta motorren pinetan 12V-etako. Balio hauek zuzenki proportzionalak direla ikusiko da. Seinale digitalaren balio altua 0 bada pultsuak ez du ezer iraungo, ondorioz, balio analogikoa 0V izango da. Seinale digitalaren balio 255 bada, seinale digitalaren balio maximoa izango da, hau da, balio altuak pultsu guztia iraungo du. Honek, 5V-eko balioa duen seinale analogikoa sortuko du. Bi limite hauen arteko erdiko pultsuk 128 balioko du, eta seinale analogikoaren balioa 2,5V-etako izango delarik. Bitarteko beste edozein balio modu berdinean kalkulatu litzateke. Komando honetan ezarritako balioaren arabera DC motorrean tentsio desberdin bat aplikatuko da. Motorraren abiadura aplikatuko tentsioaren arabera izango delarik. Laburbilduz, motorraren abiadura PWM-ari emandako balioaren arabera izango da.



*Irudia 3.3.10. PWM seinale analogikoa*

### SERBOMOTORRA ETA DRIVERRA

*MeServo servoDriver(Port\_Num,DEV\_Num)*: komando honekin serbomotorra izendatuko da eta zein portutan konektatutik dagoen adieraziko da. Motorren konexioak parentesien artean PORT\_1, PORT\_2 bakarrik izan daitezke. Portua ipintzeaz gain, serbomotorren driverrak bi serbo independenteki konektatzeko aukera eskaintzen dutenez, DEV\_1 edo DEV\_2 bietako zeinetara konektatuko den adierazi behar da.

*ServoDriver.begin(int min, int max)*: serbomotoreak biratuko duen angeluaren balio maximoa eta minimoa ezartzeko balio du.

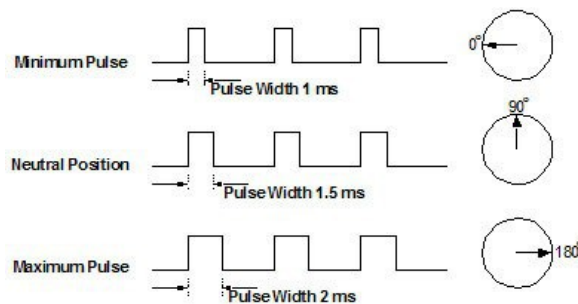
*ServoDriver.read()*: serbomotorraren ardatza zein posiziotan aurkitzen den adierazten du. Beraz, emango duen balioa 0 eta 180 bitartekoa izango da.

*ServoDriver.readMicroseconds ()*:serbomotorraren ardatza zein posiziotan aurkitzen den adierazten du. Beraz, emango duen balioa 1000 eta 2000 bitartekoa izango da.

*ServoDriver.attached ()*: komando honen bidez serbomotorea konektaturik dagoen edo ez jakingo da.

*ServoDriver.write (int value)*: Serbomotorea parentesien arteko balioa duen angelura biratuko du. Angelu honen balioa 0 eta 180-ren artekoa izan behar du.

*ServoDriver.writeMicroseconds (int value)*: serboaren ardatz biratzeko erabiltzen da. Komando hau erabilia serbomotorearen ardatzaren biraketa angelua kontrol seinaleko pultsuaren iraupenaren arabera da. Iraupen honen balioa parentesien artean ipiniko dena izango da. Balio hau 1000tik 2000rako balioa izango da. 1000 balioa duenean 0 gradutara biratuko du, 1500 ipintzean 90 gradutara kokatuko da ardatza eta 2000 balioa ipintzean 180 gradutara. Bitarteko beste edozein zenbaki puntu hauekiko zuzenki proportzionala izango da.



*Irudia 3.3.11. Biraketa angelua kontrola seinale pultsadun baten bidez*

## ULTRASOINUA

*MeUltrasonic.ultras\_sents\_izena(Port\_Num)*: komando honekin ultrasoinu sentsorea izendatuko da eta zein portutan konektatutik dagoen adieraziko da.

*ultras\_sents\_izena.distanceInch()*: ultrasoinuak detektatutako objektu edo oztopoa zenbat hazbetetara (inch) dagoen adieraziko da.

*ultras\_sents\_izena.distanceCm()*: ultrasoinuak detektatutako objektu edo oztopoa zenbat zentimetrotara dagoen adieraziko da.

## INFRAGORRIA

*MeInfraredReceiver infrg\_hartz\_izena(Port\_Num)*: komando honekin infragorri hartzailea izendatu eta zein portutan konektatutik dagoen adierazten da.

*infrg\_hartz\_izena.begin()*: Portu seriea hasierazten du.

*infrg\_hartz\_izena.available()*: Infragorri hartzailea erabilgarri dagoen adierazten du.

*infrg\_hartz\_izena.read()*: Infragorri hartzaileak seinalerik jaso duen irakurriko du.

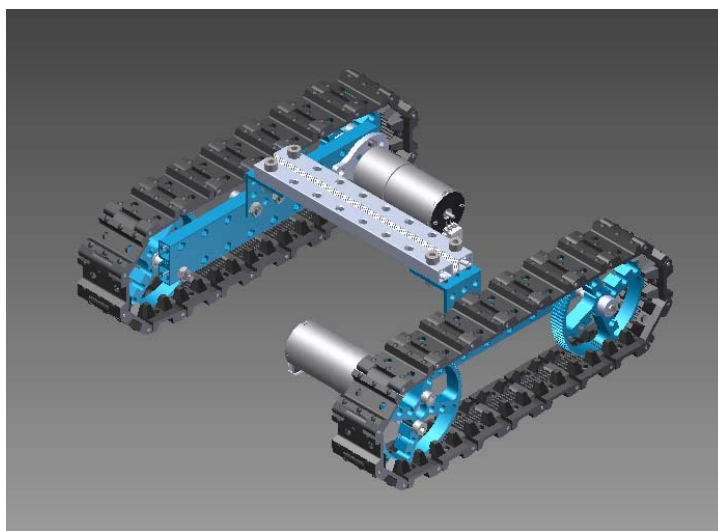
*infrg\_hartz\_izena.buttonState()*: Botoiren bat sakatuta dagoen edo ez jakiteko erabiltzen da.

## 3.4. OINARRIA

### 3.4.1. DISEINUA

Oinarria eraikitzen hasi baino lehen bere diseinua aukeratzeko ondorengo irispide hauek kontutan eduki behar dira. Alde batetik, oinarrian toki nahiko egon behar da, gero besoa gainean ipini ahal izateko. Eta beste aldetik, oinarria ahalik eta egonkorrena izatea, besoa martxan jartzen denean, robota desorekatu eta lurrera erori ez dadin. Arrazoi guzti hauengatik, egokiena oinarri beldar-kateduna erabiltzea dela erabaki da.

Beldar-katedun robotak, lurrazal mota desberdinetara moldatzeko gaitasun handia dute, hori dela eta oso robot egonkorak dira. Oinarriari mugimendua emateko binaka beldar-katearekin lotuko diren lau gurpil erabiltzen dira. Horietako bi gurpil DC motor batera konektatzen dira eta beste bi gurpilak, berriz, ardatz harilkatuetan kokatzen dira. Oinarriaren alde bakoitzean bi gurpil jartzen dira batek motorra izango duelarik eta besteak ardatz harilkatua. Baina, kontutan izan behar da, alde bateko eta besteko gurpilak kontrajarriak egon behar direla. Hau da, alde batean motordun gurpila aurrean eta bestea atzekoan ipiniko da. Izan ere, motordun bi gurpilak alde berdinean jartzen badira, oinarria ez da guztiz egonkorra izango, motorren pisu guztia alde batean kokaturik egongo delako. Horrez gain, robota garatzen ari den bitartean motorren bat askatzea beharrezkoa izatekotan, hau kontu handiarekin egin beharko litzateke biak oso hurbil egongo direlako.

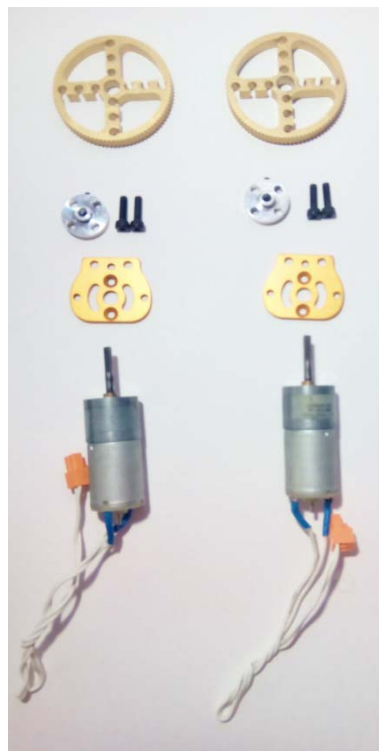


*Irudia 3.4.1. Oinarriaren 3D-ko diseinua*

### 3.4.2. MUNTAKETA

Behin oinarriaren diseinua nolakoa izango den erabakita dagoenean muntaketa prozesua hasi daiteke. Horretarako, atal honetan oinarriaren eraikuntzarako jarraitutako prozesua deskribatuko da.

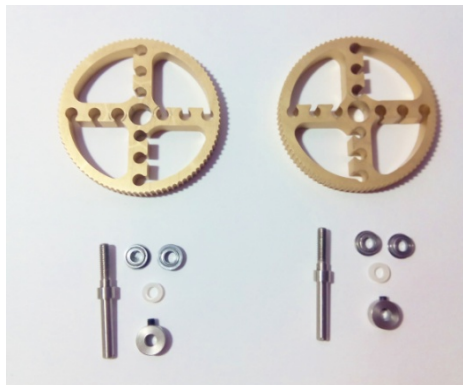
Hasteko lau gurpilak prestatu behar dira. Bi, ardatz harilkatu batean kokatzen dira eta beste biak DC motorraren ardatzean. Motordun gurpil bakoitza muntatzeko beharrezko materialak: 185rpm-tako DC motorra bat, bi M4x14 allen torloju, engranaje handi bat, finkapen torlojua, ardatz konektatzailea eta 185rpm-tako DC motorraren euskarria dira. Kasu honetan, muntaia bi urratsetan egiten da, azkeneko pauso batean guztia batzen delarik. Beraz, alde batetik, ardatz konektatzailea engranaje handiaren barruan sartzen da, kontutan izanda bi hauen zuloak parekatuta egon behar direla. Ondoren, ardatz konektoreetako zulo hariztatuak aurkitu eta zulo bakoitzetik M4x14 allen torloju bana sartu behar da. Beste aldetik, motorrean euskarria kokatzen da bi torloju txikiagoren bidez.



*Irudia 3.4.2. Motordun gurpilarentzako materiala*

Ardatz harilkatura konektatzen diren gurpil bakoitzarentzat ondoko materiala beharko da: ardatz harilkatu bat, engranaje handi bat, finkapen torlojua, finkapen eraztuna, plastikozko

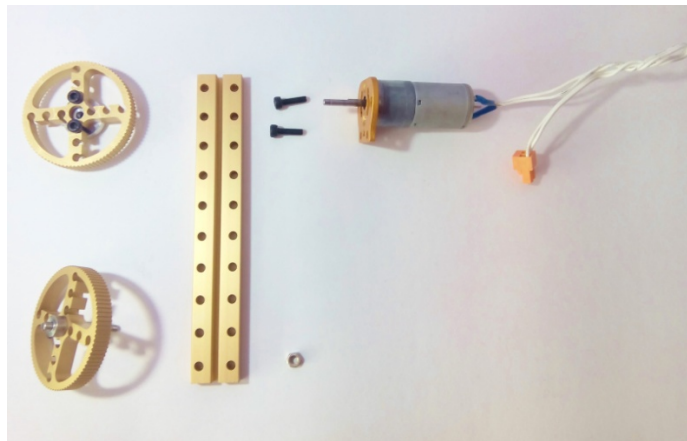
zorro banatzailea bat eta bi brida kojinete. Ardatz ez harilkatuaren aldetik honako segidan kokatuko dira piezak, lehenengo plastikozko zorro banatzailea, brida kojinete bat, engranaje handia eta beste brida kojinetea. Bukatzeko, finkapen eraztuna ipiniko da finkapen torlojuarekin lotuta guztia bere tokitik mugitu ez dadin.



*Irudia 3.4.3. Ardazdun gurpilarentzako materiala*

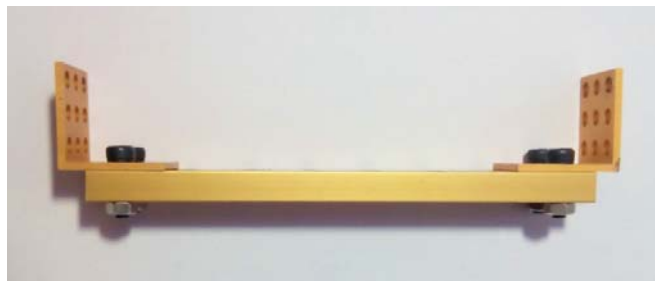
Behin gurpilen atal guztiak prest daudenean 10 zuloako bi habe hartzen dira. Habe bakoitzean motordun gurpil bat eta motor gabeko bat ipini behar dira, horretarako, azkoin bat eta bi M4x14 allen torloju erabili behar dira. Kontutan izan behar da, habe bakoitzean gurpilak alderantziz kokatuta egongo direla, batek aurrean edukiko du motordun gurpila eta besteak atzean. Ardatz harilkatua duen gurpila habeko beheko lehenengo zuloan jarriko da. Bertan finkatzeko, zuloaren alde batetik ardatza gurpilekin eta beste aldetik azkoina sartzen da. Azken hau, ardatz harilkatua estutu behar da. Motordun gurpila ordea, habearen beste aldean kokatuko da. Motorraren ardatza, habearen beheko aldean azkeneko zuloan kokatzen da. Motorra bertan finkatzeko, bi M4x14 allen torlojuak erabiltzen dira habea eta motor euskarria eusteko. Torloju bat habearen goiko zuloan eta motor euskarriaren izkinako zuloa batzen kokatzen da. Eta beste torlojua, ardatza sartu den ondoko zuloan eta euskarriaren alde lauko zuloa betetik sartzen da. Bukatzeko, hasieran prestatutako gurpila hartu eta motorraren ardatzean sartzen da. Gurpila ardatzera finkatzeko motor konektatzaileko finkapen torlojua sartu eta estutu behar da. Modu honetan, gurpilak eta motorrak batera biratuko dute. Prozesu guzti hau bi habeekin garatu behar da. Ondo eraiki dela egiaztatzeko, motordun gurpilak ez duela biratzen eta ardatz harilkatudun gurpilak biratzen duela begiratu behar da. Horrela izanez gero, oinarriaren muntaia ondo doala esan ahal izango dugu.





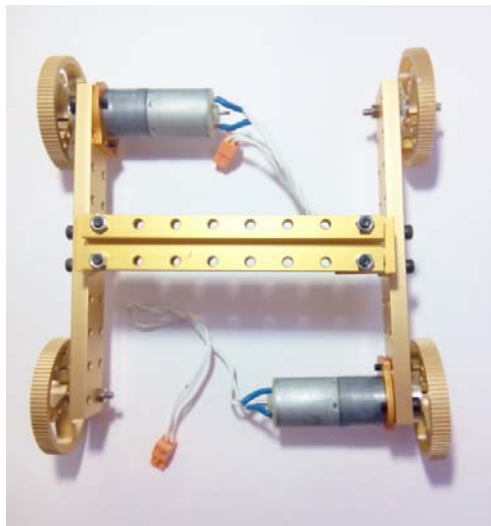
**Irudia 3.4.4.** Oinarriaren aldeentzako materiala

Oinarriaren bi aldeak eginda daudenean batu egin behar dira, beraz, hauek elkartzeko sistema eraiki behar da. Horretarako, 8 zuloko habea bat, 3x3-ko bi L xafla, lau azkoin eta lau M4x14 allen torloju erabili beharko dira. Habeari buelta ematen zaio, eta habearen mutur bakoitzean L xaflak kokatzen dira. Torlojuak, habeko muturretako zuloetan eta L xaflaren erdiko zuloetatik sartzen dira eta beste aldetik azkoinekin lotzen dira. Horrela, U itxurako egitura bat lortzen da oinarriaren alde biak elkartzeko.



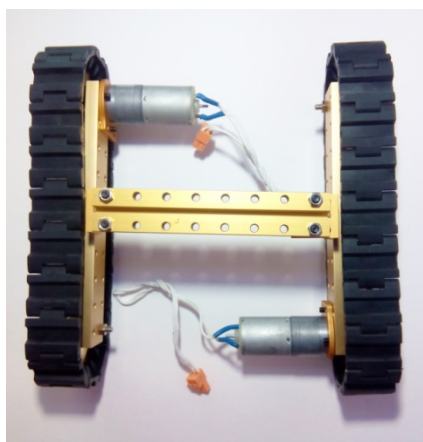
**Irudia 3.4.5.** U itxurako egitura

Ondoren, oinarriaren bi aldeak eta U egitura elkartzeko lau azkoin eta lau M4x14 allen torloju behar dira. Gurpildun bi egiturak hartu eta habearen erdian 5. eta 6. zuloetan, torlojuak sartzen dira. U itxurako sistema beherantz ipini eta torlojuak sistemaren mutur bakoitzetik sartzen dira. Horren ostean, oinarria finkatzeko torloju bakoitzean azkoin estutuko da.



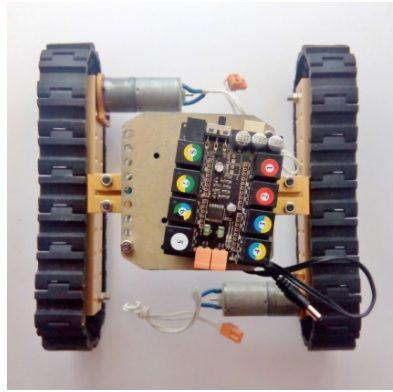
*Irudia 3.4.6. Oinarriaren egitura*

Baina hasiera batean esan den bezala, robota beldar-kateduna izatea nahi da, beraz, robotaren egiturarekin bukatzeko hau ipintzea falta da. Bi beldar-kateak hartu eta bakoitza alde batean ipini behar da. Beldar katea jartzeko modurik errazena, lehenengo ardatz harilkatuko engranajeetan sartzen da eta ondoren bestean. Berdina egin beharko da bestearekin.



*Irudia 3.4.7. Oinarriaren egitura osorik*

Bukatzeko, robotari mugimendua emateko Arduino UNO eta MAKEBLOCK-eko plaka eta hauek elikatzeko beharrezko diren pilak kokatu beharko dira. Guzti hau, habearen gainean ipini ahal izateko plaka euskarria erabiliko da. Pila kaxa bere euskarriarekin kokatzen da habearen gainean eta bi azkoin eta bi M4x14 allen torlojuren bidez lotzen da. Ondoren, pilen gainean plaka euskarria kokatzen da. Beste bi azkoin eta bi M4x22 allen torlojuren bidez pilen euskarria lotzen da. Torlojuok, euskarri batetik bestera justu heltzen direnez, kontu handiarekin ibili behar da.



*Irudia 3.4.8. Oinarria muntatuta plakekin*

Oinarria guztia muntatu dagoenean, diseinu atalean ezarritako irispide guztiak bete direla egiaztatuko da. Honekin, beldar-katedun robotaren oinarria eraikita geratzen da eta goiko aldean beso robotikoa kokatu ahal izateko toki nahiko dagoela ere ziurtatuta geratuko da. Momentuz oinarriaren goiko aldean habe bakararra dago. Habe horren gainean robotaren oinarria programatzeko Arduino plaka eta MAKEBLOCK-eko plaka kokatzeko nahikoa delako. Baina aurrerago, beso robotikoa ipintzerako orduan, beharrezkoa bada habe gehiago kokatzeko aukera egongo da.

### **3.4.3. PROGRAMAZIOA**

Atal honetan zehar, robotaren oinarriaren programazioaren garapena azalduko da. Lehendabizi, oinarrizko programak garatzen dira, ondoren beharrezko osagai eta programa funtzioak gehitzen joango dira, programa nagusia guztiz osatuta dagoen arte. Robota programatzeko Arduino softwarea erabiliko da. Gainera, MAKEBLOCK-eko osagai elektronikoak erabiltzen direnez, MAKEBLOCK-eko web orri ofizialean aurkitzen den Arduino-rentzako MAKEBLOCK-eko liburutegia erabiliko da. Beraz, Arduino software eta MAKEBLOCK-eko liburutegia erabilita osagai elektronikoak programatuko dira, robota higikortasuna emateko. Liburutegi honen bitartez osagai elektroniko bakoitza programatzeko komando aurkituko dira.

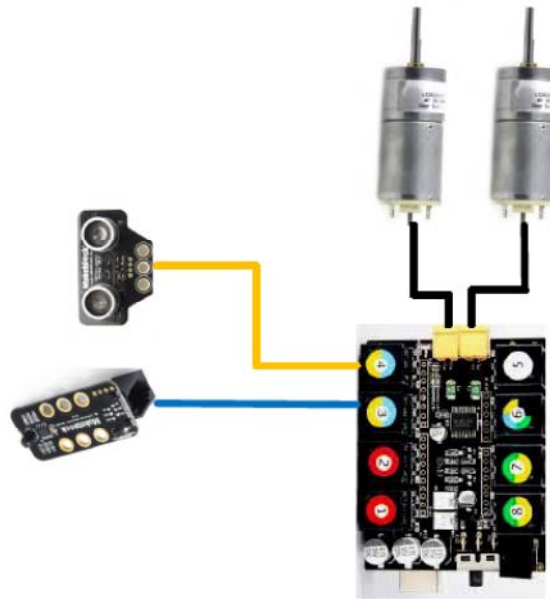
Programak idazten hasi baino lehen, oinarriaren programa nagusiaren helburua zehaztu egin behar da, hau da, robotaren oinarriak zer egingo duen erabaki behar da. Oinarria eta besoen mugimenduak gidatzeko, mando infragorri erabiliko da, horrela, robotak erabiltzaileak agindutakoa egingo du. Guzti hau lortzeko, beharrezko materiala hartzailea

infragorri eta mando infragorria dira. Erabiltzaileak robota kontrolatzeaz gain, oinarria oztopo eta paretak ekiditeko programatuko da. Modu honetan, robotak kolpe gutxiago hartuko ditu eta gehiago iraungo du hautsi gabe. Azken helburu hau lortzeko ultrasoinu sentsorea erabiliko da. Ultrasoinuaren bidez, aurrean oztoporen bat dagoen eta zein distantziatara aurkitzen den detektatuko du.

Programatzen hasi orduko osagai elektroniko guztiak plakara konektatuta egongo dira, nahiz eta hasiera batean ez erabili, horrela behar direnerako konektaturik egon dira. Osagai elektronikoak nola konektatu behar diren jakiteko, osagai bakoitzak sarreran, RJ konexioaren gainean, daukan kolorea begiratu behar da. Ondoren, MAKEBLOCK-eko plakan begiratu gero, sarrera/irteera bakoitzak kolore bat edo bat baino gehiago dituela ikusiko da. Pin horietara kolore bera duen edozein osagai elektroniko konektatu daiteke. Esate baterako, infragorri hartzailearen kolorea urdina da, hortaz, MAKEBLOCK-eko plakan kolore urdina duen edozein sarrera/irteerara konektatu beharko litzateke. MAKEBLOCK-eko plakan pin batek baino gehiagok dute kolore urdina, orduan, horietako edozeinetara konektatu daiteke.

Behin konexioa pin batera egitea aukeratu denean hobe da hau ez mugitzea, programak idazterakoan osagai elektronikoak zein sarrera/irteerara konektatuta dauden adierazi behar delako. Horrek esan nahi du, konexio pina aldatzen ibiltzen bada, programan informazio hori aldatu behar denez, errazago izango litzateke kodean erroreak eta nahasteak agertzea. Esaterako, garatutako programan osagai elektroniko bat sarrera/irteera batera konektatuta dagoela ipintzea eta horrela ez izatea, honek oinarriaren programak ez funtzionatzea ekarriko luke.

Kolore kodeari begiratu ostean, proiektuan zehar konexio hauek erabiliko dira erabaki da: Hartzaile infragorria 3. pinera eta ultrasoinu sentsorea 4. pinera konektatuko da. MAKEBLOCK-eko plaka osagai elektronikoak konektatzeaz gain, motorrak ere konektatu behar dira. Motorrak plakaren alde batean aurkitzen diren M1 eta M2 portuetara konektatuko dira. Guzti hauek oinarriak programatzeko behar diren konexio guztiak dira 17. Irudian argiago ikus daitezke.



**Irudia 3.4.9.** Oinarriko konexioak

Behin programaren helburua erabakita eta finkatuta dagoenean eta osagai elektroniko guztiak plakara konektatuta daudenean guztia prest dago robotaren oinarria programatzen hasteko. Oinarriaren programa nagusia lortzeko prozesurik egokiena, programa sinpleenetatik hastea da, hau da, lehendabizi motorrak bakarrik aktibatuz. Ondoren, banan banan osagai elektronikoak eta beharrezkoak diren beste funtzioak kodetzen dira programazio zailduz, ezarritako helburura iritsi arte. Programa bakoitza egin ahala, izendatu behar da beraien artean nahasterik ez sortzeko.

Arduino programazio inguruaren bidez garatuko den lehenengo programa *oinarria1* da, honen bitartez oinarriaren eraikuntza eta osagai elektronikoaren funtzionamendua zuzena dela egiaztatuko da. Horrez gain, MAKEBLOCK liburutegiarekin lehenengo kontaktua egingo da, DC motorrak programazioa landuz, hau da, motor hauek nola izendatu, biratu edo nola geldituko den ikusiko delarik. Programa honetan ondorengo aginduak kodetzen dira sekuentzialki exekutatu daitezkeen: motorrek alde batera biratuko dute 3 segundoz, motorrak geldirik egongo dira 1 segundoz, motorrek beste aldera biratuko dute 3 segundoz eta bukatzeko motorrak geldirik egongo dira 6 segundoz. Motorren biraketa hauekin, hasiera batean, robotaren oinarriak aurrerantz mugituko lerro zuzenean. Ondoren, gelditu egingo da eta segituan lerro zuzenean atzerantz joango da. Bukatzeko, segundo batzuek geldirik egongo da, programa, berriz, exekutatzen hasi arte.

```

/*****
Programaren izena: oinarrial
Egilea: Itziar Aldekoa
Deskribapena: programa honetan oinarria aurrera eta atzera mugituko
*****/

#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeDCMotor m_esk(M1); // M1 sarrera eskumako motorrarena
MeDCMotor m_ezk(M2); // M2 sarrera ezkerreko motorrarena

byte motor_abid = 150; // motorren abiadura, PWM-a balio maximoa

void setup()
{

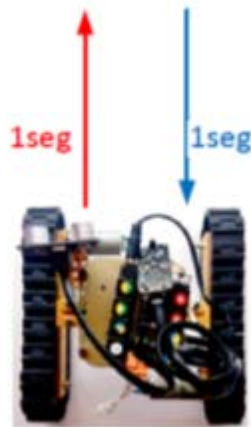
}

void loop()
{
  m_esk.run(motor_abid); // motorrak biraka, AURRERA MUGITU
  m_ezk.run(motor_abid);
  delay(3000); // 3 segunduz
  m_esk.stop(); //motorrak GELDIRIK
  m_ezk.stop();
  delay(1000); // 1 segunduz
  m_esk.run(-motor_abid); // motorrak biraka (beste aldera), ATZER
  m_ezk.run(-motor_abid);
  delay(3000); // 3 segunduz
  m_esk.stop(); //motorrak GELDIRIK
  m_ezk.stop();
  delay(6000); // 3 segunduz
}

```

*Irudia 3.4.10. oinarrial programa*

Robotaren oinarriaren funtzionamendu zuzena ziurtatzeko lehenengo zeregina oinarriaren mugimenduak behatzea da. Hauek lehen deskribatu diren bezala (3.4.10 Irudian adierazten den moduan) egiten ez badira, funtzionamendu okerra gauzatu duen arazoa identifikatu eta konpondu behar da. Funtzionamendu okerra modu desberdinetan agertu daiteke.



*Irudia 3.4.11. Oinarrial  
exekutatzean egingo duen ibilbidea*

Alde batetik, baliteke oinarria ez mugitzea, hau arrazoi desberdinengatik izan daiteke, ohikoenak: plaka ez piztea, pilak MAKEBLOCK plakara konektatu beharrean Arduino plakara konektatu izana, pilak agorturik izatea edo bestela baliteke programa kargatzerakoan ez konturatzea eta errore bat eman izana. Hori dela eta, oinarria mugitzen ez bada lehenengo egin beharko diren gauzak, plakako etengailua ON egoeran dagoela, pilak MAKEBLOCK-eko plakara konektaturik daudela, pilak gastaturik ez daudela eta programa plaka kargatu dela ziurtatzea da.

Bestetik, baliteke motor batek ez biratzea (besteak biratzen duenean), orduan, baliteke torlojuren bat motorrera lotzean larregi estutu izana, eta horregatik, motorrak ez biratzea. Arazo hau antzematea eta konpontzea aurrekoak baino zailagoa da, horregatik, hobe da lehendabizi beste arazo guztiak begiratzea. Bigarren arazo hau konpontzeko, ez dabilen motorra oinarritik askatuko da eta funtzionatzen duela ziurtatu. Ondoren, inguruko torloju guztiak konprobatu behar dira baten batek arazorik ematen duen jakiteko, horrela bada, torlojua pixka bat askatzea nahiko izango litzateke.

Oinarria martxan jartzerakoan aurkitu daitekeen beste arazo bat, oinarriaren mugimendua lerro zuzenean aurrerantz eta antzerantz joan beharrean mugimenduak norabide okertua izatea da. Programatutako kodearen arabera, arazo hau identifikatzeko jakina da oinarria mugitzen hasten den leku berdinean bukatu beharko lukeela, horrela gertatzen ez bada oinarria okertzen delako da. Oinarria okertzen bada, alde bateko motorrak besteak baino astiroago biratzen duelako izango da. Oinarria desbideratzen den aldeko motorra izango da

arazoa ematen duena. Arrazoi desberdinengatik izan daiteke, baliteke lehen torlojuak askatzerakoan behar beste ez askatu izana edo baliteke motor bat hondaturik egotea eta beste baino geldoago ibiltzea, horrela izanez gero, hondatuta dagoen motorra aldatu egin beharko litzateke.

Laburbilduz, Arduino programazio ingurunean garatu *oinarria1* programa bi eginkizun betetzea lortu da. Alde batetik MAKEBLOCK-eko liburutegiarekin lehenengo kontaktua izatea, DC motorren programazioak garatuz eta bestetik oinarriaren muntaia egokia dela ziurtatzea. Lehen aipatu den bezala, torloju bat larregi estutzeak motorra ez funtzionatzea edo txarto funtzionatzea ekar dezake. Edo motor bat hondatua egotea oinarriaren mugimendu zuzena okertzea dakar. Beraz, oso garrantzitsua izan da lehenengo programa hau garatu izana, zuzenean programazio nagusia eginez gero txarto funtzionatuko balu, zailago izango litzateke arazoa aurkitzea. Modu honetan, hasieratik guztiak ondo funtzionatzen duela ziurtatzen da, aurrerago arazoak ekidituz.

Garatuko den bigarren programa *oinarria2* izango da. Hau aurrekoaren antzekoa izango da, baina lehen oinarriari aurrerako eta atzerako mugimendua eman zitzaion bitartean, oraingoan, oinarriari eskumara eta ezkerrera biratzeko gaitasuna emango zaio. Horrela, robotaren oinarriak izan ditzakeen mugimendu guztiak aztertuko eta probatuko dira. *oinarria1* programatik abiatuz *oinarria2* programa kodetuko da. *Oinarria2* programan aurrera eta atzerako mugimenduen artean, lehenengo 3 segundotan zehar eskumara eta ondoren beste 3 segundoz ezkerrera biratzeko aginduko zaio. *Oinarria2* programan, oinarriaren mugimendu bakoitzaren artean 0,2 segundotako geldialdiak egingo dira, modu horretan motorrek aldaketa egiteko denbora izango dute kolperik jo gabe eta oinarria ez da denbora luzez geldirik egongo, horri esker mugimenduak ia jarraian egiten direla irudituko da.



*Irudia 3.4.12. Oinarria2 exekutatzean egingo duen ibilbidea*



Oinarriaren biraketa programatzeko bi modu daude: motor bakoitzak alde batera biratzea edo motor bat geldirik eta bestea martxan egotea. Lehenengo kasuan, motorrak bere ardatzaren gainean biratzen du, bigarren kasuan, ordea, oinarriaren biraketa desplazamenduarekin batera ematen da, hori izango litzateke bi biraketa hauen arteko desberdintasunik nabarmenena. Hala ere, kasu bietan alde berdinerira biratuko luke, kontrako norabidean biratzen duen motorraren aldera edo geldik dagoen motorraren aldera. *Oinarria2* programa egiterakoan bi kasuak aztertu dira, oinarria nola mugitzen den ikusteko, eta proiektua garatzeko, azkenean, biraketa oinarria bere ardatzaren gainean biratuko duela erabaki da. Horrela, urruneko aginduak ematerako orduan hobeto kontrolatuko delakoan.

```

/*****
Programaren izena: oinarria2
Egilea: Itziar Aldekoa
Deskribapena: programa honetan oinarria aurrera mugitu, gero eskum
ondoren ezkerretera biratuko du eta bukatzeko atzera mugituko da
*****/

#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeDCMotor m_esk(M1); // M1 sarrera eskumako motorrarena
MeDCMotor m_ezk(M2); // M2 sarrera ezkerreko motorrarena

byte motor_abid = 150; // motorren abiadura, PWM-a ba

void setup()
{

}

void loop()
{
    //AURRERA MUGITU
    m_esk.run(motor_abid); // motorrak biraka motor_abid-rarek
    m_ezk.run(motor_abid);
    delay(3000); // 3 seg
    //GELDITU
    m_esk.stop(); //motorrak geldirik,
    m_ezk.stop();
    delay(200); // 0,2 seg
    //ESKUMARA BIRATU robotaren ardatzaren gainean
    m_esk.run(-motor_abid); //motor bakoitza alde batera biratu
    m_ezk.run(motor_abid);
    // BESTE AUKERA eskumara biratzeko
    //m_esk.stop(); robotak biratzen trasladatzen den bitartean

```

*Irudia 3.4.13. oinarria2 programa (1.zatia)*

```

//m_ezk.run(motor_abid); // ezker motor. biraka eta esk geld
delay(3000); // 3 seg
//GELDITU
m_esk.stop(); //motorrak biak geldirik,
m_ezk.stop();
delay(200); // 0,2 seg
//EZKERRERA BIRATU robotaren ardatzaren gainean
m_esk.run(motor_abid); //motor bakoitza alde batera biratu
m_ezk.run(-motor_abid);
// BESTE AUKERA ezkerrera biratzeko
//m_esk.run(motor_abid); robotak biratzen trasladatzen den!
//m_ezk.stop(); // eskumako motor. biraka eta ezkerrekoa ge
delay(3000); // 3 seg
//GELDITU
m_esk.stop(); //motorrak biak geldirik,
m_ezk.stop();
delay(200); // 0,2 seg
// ATZERA MUGITU
m_esk.run(-motor_abid); //motorrak biraka beste aldera,
m_ezk.run(-motor_abid);
delay(3000); // 3 seg
//GELDITU
m_esk.stop(); //motor biak geldirik,
m_ezk.stop();
delay(6000); // 6 seg
}

```

*Irudia 3.4. 14. oinarria2 programa (2.zatia)*

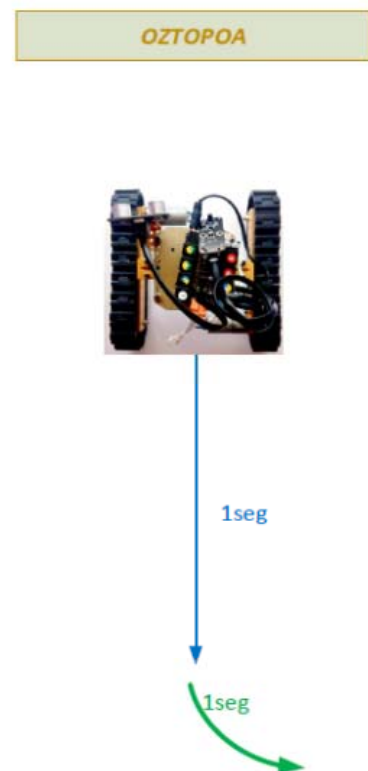
*Oinarria2* programaren bidez kodetutako oinarriaren funtzionamendua egiaztatzerakoan agertu daitekeen arazoa oinarriak biratzerakoan motorren batean kokaturiko engranajea eta beldar-katea ardatzetik askatzea da. Honen konponbidea, ardatz konektatzaileko finkapen torlojua estutzean datza.

Laburbilduz, *oinarria2* programan, oinarriak egin ahal izango dituen mugimendu guztiak zehaztu dira. Are gehiago, oinarriaren biraketa modua bere ardatzaren gainean izango dela ere zehaztu egin da. Lehen esan bezala, ardatzaren gaineko biraketaren programa garatzen denean, motor bakoitzak alde batera biratuko duela, eta gogoan izan behar da oinarriak kontrako noranzkoan biratzen duen alderantz biratuko duela.

Ondorengo programetan zehar, orain arte erabili ez diren osagai elektronikoak beharko dira. Honek esan nahi du, apurka programak zailtzen joango direla eta kontu handiagoarekin ibili beharko dela, ahalik eta akats gutxien egiteko. Lehendabizi, ultrasoinu sentsorearen

funtzionamendua programatuko da, oinarria ez dadin ormen edo oztopoen kontra jo. Ondoren, infragorri hartzaile eta aginte infragorria oinarria gidatzeko erabili ahal izateko programak garatuko dira. Eta azkenik biak bateratuko dira, biekin programa bakarra sortuz. Hau da, erabiltzaileak aginte infragorriarekin kontrolatuko du oinarria, baina, oinarriak oztopo edo objektu bat detektatzen duenean bertatik aldentuko da.

Ultrasoinu sentsorea kontrolatzen duen programari *ultrasoinua* izena ezarri zaio. Atal honen hasieran esan bezala, sentsore hau MAKEBLOCK plakako 4. pinera konektatuta dagoela kontutan izan behar da programa eraikitzeke. Programa honen helburua, ultrasoinu sentsorearen programazioa eta funtzionamendu zuzena lortzea da. *ultrasoinua* programaren eginkizuna, oinarria oztopoengandik aldentzea da, oinarria ezeren kontra kolpatu ez dadin. Hau da, oinarria ultrasoinuarekin kudeaturiko erreakzio-erantzunaren bidezko gidatzea garatuko da. Kodea eraikitzerakoan, ultrasoinu sentsoreak 15cm-tara edo gertuagora oztopoak aurkitzeko programatuko da. Honen ondorioz, ultrasoinuak 15cm baino urrunago dauden objektuak detektatzean oinarriak aurrerantz zuzen jarraituko du, baina, ultrasoinuak objektu bat 15cm edo gutxiagotara dagoenean detektatzen duenean oztopotik aldentuko da. Oinarriaren erantzunaren programazioan oztopoa topatzen duenean: lehenengo, motorrak geldirik egongo dira 0,2 segundoz, ondoren, motorrak aurkako norabidean biratuko dute 1 segundoz, oinarria atzerantz mugituz. Azkenik, oinarriak bere ardatzaren gainean eskumara biratuko du 1 segundoz, horretarako, eskumako motorrak aurkako norabidean biratuko du.



**Irudia 3.4. 15.** Oinarria2 exekutatzean egingo duen ibilbidea

```

/*****
Programaren izena: ultrasoinua
Egilea: Itziar Aldekoa
Deskribapena: Ultrasoinua probatzeko erabiliko den programa simple
da. Oinarria aurrerantza joango da eta ultrasoinuak 15cm zerbait
detektatzen duenean beregandik urrunduko da, atzera eginez eta gero
pixkat biratuz
*****/

#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeDCMotor m_esk(M1); // M1 sarrera eskumako motorrarena
MeDCMotor m_ezk(M2); // M2 sarrera ezkerreko motorrarena
MeUltrasonicSensor ultras_sents(PORT_4); // ultrasoinua 4. portura

byte motor_abid = 150; // motorren abiadura, PWM-a balio maximoa

void setup()
{

}

void loop()
{
  if (ultras_sents.distanceCm() > 15) // 15cm baino handiagoko dist.
  {
    m_esk.run(motor_abid); // AURRERA MUGITU,
    m_ezk.run(motor_abid); //motorrak biraka motor_abid-rarekin
    delay(100); // 1 seg
  }
  else
  {
    m_esk.stop(); // GEKDIRIK, Motorrak gelditu
    m_ezk.stop();

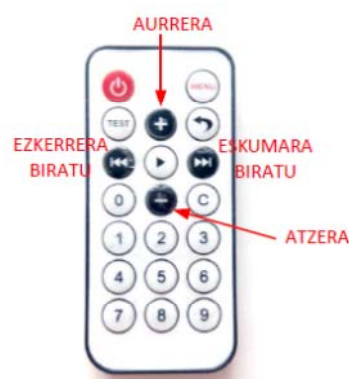
    delay(200); //0,2 seg
    m_esk.run(-motor_abid); // ATZERA MUGITU, motorrak biraka
    m_ezk.run(-motor_abid); //kontrako norabidean
    delay(1000); // 1 seg
    m_esk.run(-motor_abid); // ESKUMARA BIRATU ardatzaren gainean,
    m_ezk.run(motor_abid); //horretarako motor bakoitza alde bater
    delay(1000); // 1 seg
  }
}
}

```

*Irudia 3.4. 16. Ultrasoinu programa*

Programa robotaren oinarrian konprobatzerakoan, baliteke ultrasoinuaren erreakzio denbora handiegia izatea. Ondorioz, oinarria horma edo oztoporen baten kontra kolpatzea da eta handik gutxira bertatik urruntzea. Arazo hau konpontzeko, if-aren barruan aurkitzen den Delay-ra jo behar da. Programazio inguruneko atalean azaldu den bezala, programazioan delay bat ipiniz gero programa gelditu egiten da, beraz, denbora horretan zehar ez du ultrasoinuak distantziarik neurtzen. Horregatik, delay-ak irauten duen bitartean, oinarriak aurrerantz jarraitzen du horma edo oztopoaren kontra jotzea delarik. Eta handik gutxira, delay-tik irteten denean, oinarria gelditu eta aldenduko da, izan ere, delay-tik irteteen ultrasoinuak distantzia neurtuko du. Beraz, denbora horren balioa txikitu egin beharko da. Denbora hori zehazteko modurik egokiena probak egiten joatea da. Erantzuna nahiko azkarra denean, hau da, ez dela ezeren kontra joko ziurtatzean, denbora horretan itxiko da. Modu honetan zehaztu da proiektu honetan erabiliko den denboraren balioa 0,1 segundotakoa izatea.

Laburbilduz, *ultrasoinua* programari esker, ultrasoinu sentsorea programatzen ikasi da, honen espezifikazio desberdinak nola zehaztu behar diren ikusi da, osagai elektroniko honek MAKEBLOCK liburutegian dituen komandoak ezagutza hobetuz. Horrez gain, programa honi esker oinarriak automatikoki mugitzeko gaitasuna izango du, talkak saihestuz eta oinarriaren erreakzio denboraren balioa zehaztuz.



*Irudia 3.4. 17. Aginteko kontrola*

Orain arte, oinarriaren mugimendua automatikoa izan da, programaren aginduak bata bestearen ondoren, sekuentzialki, exekutatzen zirelarik. Hurrengo programaren bidez, *oinarria\_infrag* programan, infragorria hartzaile bat eta agente infragorria erabiliko dira oinarriaren mugimendua erabiltzaile baten aginduen araberakoa izan dadin. Lehenago azaldu

den bezala, infragorri hartzailea MAKEBLOCK-eko plakaren 3.pinean konektatu behar da. Aginte infragorria, ordea, ez da inora konektatu behar, pila duela ziurtatu besterik ez da egin behar bere funtzionamendu zuzena ziurtatzeko. Agintean dauden goi aldeko botoiekin oinarria zein norabidetan joan behar den esaten zaio. Gehi eta ken botoiei ematerakoan oinarria aurrera eta atzera mugituko da. Eta geziekin eskumara edo ezkerredera bakoitzak adierazten duen alderantz oinarria buruaren gainean biraraziko dute.

```

/*****
Progrmaren izena: oinarria_infrag
Egilea: Itziar Aldekoa
Deskribapena: Oinarria, mando infragorriaren bidez kontrolatuko da.
+ eta - botoiekin oinarria aurrera eta atzera mugituko da. <-- boto
ezkerredera eta --> eskumara biratuko du
*****/
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeDCMotor m_esk(M1); // M1 sarrera eskumako motorrarena
MeDCMotor m_ezk(M2); // M2 sarrera ezkerreko motorrarena
MeInfraredReceiverinfragorri_hartz(PORT_3);
//hartzaile infragorria 3.pinera konektatu

int motor_abid = 190; // motorren abiadura, PWM-a balio maximoa 25

void setup()
{
  infragorri_hartz.begin();
}

void loop()
{
  if(infragorri_hartz.available() || infragorri_hartz.buttonState())
  {
    switch(infragorri_hartz.read())
    {
      case IR_BUTTON_PLUS: // mandoko + botoia aurrera mugitu
        Aurrera();
        break;
      case IR_BUTTON_MINUS: // mandoko - botoia atzera mugitu
        Atzera();
        break;
      case IR_BUTTON_NEXT: // mandoko --> botoia eskumara birat
        Eskumara();
    }
  }
}

```

*Irudia 3.4.18. oinarria\_infrag programa (1.zatia)*

```

        break;
    case IR_BUTTON_PREVIOUS:// mandoko <-- botoia ezkerrera b
        Ezkerrera();
        break;
    }
}
else
{
    Stop();
}
}

//Goian izendatutako botoiari sakatzean exekutatuko diren programa
void Aurrera()// AURRERA MUGITU, motor biak biratzen
{
    m_esk.run(motor_abid);
    m_ezk.run(motor_abid);
}

void Atzera()// ATZERA MUGITU, motor biak beste aldera biratzen
{
    m_esk.run(-motor_abid);
    m_ezk.run(-motor_abid);
}

void Ezkerrera()// EZKERRERA BIRATU, motor bakoitza alde batera bi
// ezkerrekoa kontrako norabidean
{
    m_esk.run(motor_abid);
    m_ezk.run(-motor_abid);
}

void Eskumara()// ESKUMARA BIRATU, motor bakoitza alde batera bira
// eskumakoa kontrako norabidean
{
    m_esk.run(-motor_abid);
    m_ezk.run(motor_abid);
}

void Stop()// GELDIRIK, motor biak geldik
{
    m_esk.stop();
    m_ezk.stop();
}

```

*Irudia 3.4. 19. oinarria\_infrag programa (2.zatia)*

Oraingoan, oinarriaren funtzionamendua probatzerakoan kontutan izan behar da oinarria lurrean ipintzean ez dela mugitzen hasiko urrutiko aginteko botoiren bati eman arte. Urrutiko aginteak ondo funtzionatzeko, botoiren bati ematean agintea oinarrirantz zuzendu

behar da eta hartzaile infragorria agindu seinalea ahalik eta hoberen jaso dezan oinarriaren goiko aldean kokatuta egon behar da. Gainera, infragorriez kontrolatutako beste edozein dispositiboan gertatzen den bezala, bien artean ezin da oztoporik egon, bestela, argi infragorria ez da hartzailera iritsiko.

Laburbilduz, MAKEBLOCK-eko liburutegia erabilia beste osagai elektronikoa bat, hartzaile infragorriaren programazioa nola garatu ikasi da, horrez gain, orain arte erabili gabeko funtzioak erabili dira. Bestetik, hasieran ezarritako helburua lortu da, agente infragorria eta hartzaile infragorriaren bidezko oinarriaren kontrola garatu delarik.

Azkenik, oinarriaren ultrasoinu sentsore bidezko mugimendu automatikoa eta urrutiko agentearen bidez kontrolatutako mugimendua *oin\_infrag\_ultras\_5* programan batuko dira. Programa honekin, oinarria urrutiko agentearen bidez gidatuko da, baina, oztoporen batekin aurkitzen denean bertatik urrundu beharko da, nahiz eta, erabiltzaileak oztopoaren kontra jotzeko agindua eman. Horregatik da hain garrantzitsua aurreko bien funtzionamendu egokia aurretik egiaztatzea. Gainera, zatikako garapen honi esker, osagai bakoitzaren programazioa ezagututa garatu ahal izango da programa berria. Bi funtzionamendu moduak erabiliko direnez lehenengo argi eduki behar da bietatik zein agindu den nagusiena, erabiltzailearena edo oztopoa ekiditzearena. Beraz, programa honen funtzionamendua azaltzerakoan esan bezala, nahiz eta erabiltzaileak oinarriari aurrerantz joateko esan, honek zerbait dagoela antzematen duenean bertatik aldentuko da, agente infragorriari kasurik egin gabe. Azalpen honen bitartez argi ikusten da, ultrasoinu sentsorea garrantzitsuena dela eta hori kontutan hartuko da programa garatzerakoan. Programazioa bitan zatitzen da. Alde batetik ultrasoinuak neurturiko distantzia 15cm baino handiagoa denean, orduan oinarriak agente infragorriak esandakoa egingo duen zatia. Beste aldetik, ultrasoinuak neurturiko distantzia 15cm baino txikiagoa denean, nahiz eta, erabiltzaileak agente infragorriari eman oinarriak oztopoa baten aurrean dagoela antzemango du eta bertatik urrunduko da agente infragorriari kasurik egin gabe.



```

/*****
Programaren izena: oin_infrag_ultras_5
Egilea: Itziar Aldekoa
Deskribapena: Oinarria, aginte infragorriarekin gidatuko da, baina,
ultrasoinuak 15cm edo gutxiagotara zerbait badago bertatik alenduko da,
nahiz eta, erabiltzaileak aurrera joateko agindua eman.
*****/
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeDCMotor m_esk(M1); // M1 sarrera eskumako motorrarena izango da
MeDCMotor m_ezk(M2); // M2 sarrera ezkerreko motorrarena izango da
MeInfraredReceiver infragorri_hartz(PORT_3);
// infragorri hartzaila 3.portuan konektatu
MeUltrasonicSensor ultras_sents(PORT_4); // ultrasoinua 4. portura konektatu

byte motor_abid = 190; // motorren abiadura, PWM-a balio maximoa 255

void setup()
{
    infragorri_hartz.begin();
}

void loop()
{

    if((infragorri_hartz.available() || infragorri_hartz.buttonState()) && (ultras_sents.distanceC
    {
        switch(infragorri_hartz.read())
        {
            case IR_BUTTON_PLUS: // mandoko + botoia aurrera mugitu
                Aurrera();
                break;
            case IR_BUTTON_MINUS: // mandoko - botoia atzera mugitu
                Atzera();
                break;
            case IR_BUTTON_NEXT: // mandoko --> botoia eskumara biratu
                Eskumara();
                break;
            case IR_BUTTON_PREVIOUS: // mandoko <-- botoia ezkerreko biratu
                Ezkerreko();
                break;
        }
    }
    else if ((infragorri_hartz.available() || infragorri_hartz.buttonState()) && (ultras_sents dist
    {
        m_esk.stop(); // GEKDIRIK, Motorrak gelditu
        m_ezk.stop();
        delay(200); // 0,2 seg
        m_esk.run(-motor_abid); // ATZERA MUGITU, motorrak biraka
        m_ezk.run(-motor_abid); // kontrako norabidean
        delay(1000); // 1 seg
        m_esk.run(-motor_abid); // ESKUMARA BIRATU ardatzaren gainean,

```

*Irudia 3.4.20. oin\_infrag\_ultras5 programa (1.zatia)*

```

        m_ezk.run(motor_abid); //horretarako motor bakoitza alde batera
        delay(1000); // 1 seg
    }
    else
    {
        Stop();
    }
}

//Goian izendatutako botoiari sakatzean exekutatuko diren programak
void Aurrera()// AURRERA MUGITU, motor biak biratzen
{
    m_esk.run(motor_abid);
    m_ezk.run(motor_abid);
}

void Atzera()// ATZERA MUGITU, motor biak beste aldera biratzen
{
    m_esk.run(-motor_abid);
    m_ezk.run(-motor_abid);
}

void Ezkerrera()// EZKERRERA BIRATU, motor bakoitza alde batera biratu
{
    // ezkerrekoa kontrako norabidean
    m_esk.run(motor_abid);
    m_ezk.run(-motor_abid);
}

void Eskumara()// ESKUMARA BIRATU, motor bakoitza alde batera biratu
{
    // eskumakoa kontrako norabidean
    m_esk.run(-motor_abid);
    m_ezk.run(motor_abid);
}

void Stop()// GELDIRIK, motor biak geldik
{
    m_esk.stop();
    m_ezk.stop();
}

```

*Irudia 3.4.21. oin\_infrag\_ultras5 programa (2.zatia)*

Laburbilduz, azken programa honekin oinarriaren funtzionamendu garatu da. *oin\_infrag\_ultras\_5* programa eraikitzeke oso erabilgarriak izan dira aurretiaz garaturiko programa guztiak. Izan ere, programa hauei esker, osagai elektronikoen ondo funtzionatu dute eta MAKEBLOCK-eko liburutegia erabilia osagai bakoitzaren kodeketa ezagunak izan dira. Guzti honi esker, hasieran ezarritako helburua bete da, erabiltzaileak oinarria kontrolatzen du aginte infragorri bidez, baina aurrean oztoporen bat aurkitzen badu bertatik aldentzeko da erabiltzaileari kasurik egin gabe.

## 3.5. BESOA

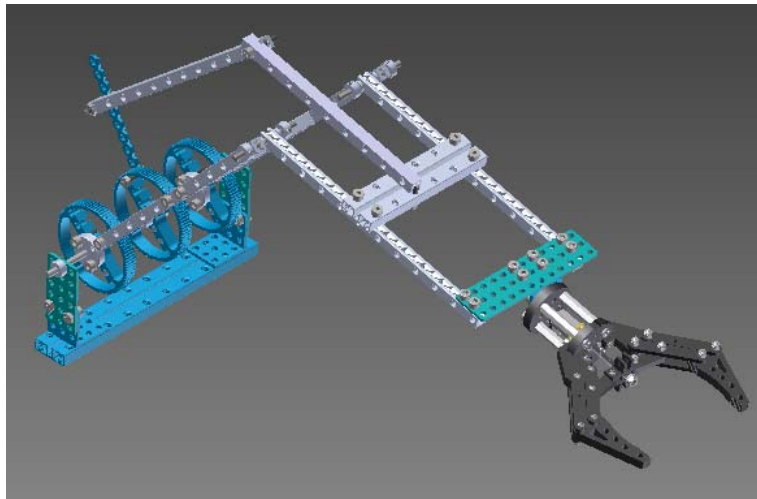
Alternatibak aztertu diren atalean esan bezala, bi beso desberdin garatu dira bata beso robotiko tradizionala eta bestea transmisio zuzeneko beso robotikoa. Hori dela eta, diseinu, muntaketa eta programazio ataletan bi besoen garapena deskribatuko da, honek biak hobeto alderatzeko aukera eskainiko du, besoen puntu komunak eta desberdintasunak argiago adierazteko asmoarekin.

### 3.5.1. DISEINUA

Diseinuari dagokionez, bi besoek 3 askatasun gradu izango dituzte. Bi askatasun gradu besoen artikulazioetan egongo dira. Artikulazioen biraketak, kate-mailak gora eta behera higiarazten du. Hirugarren askatasun gradua, besoen bukaeran kokaturiko pintzan emanda etorriko da, pintza ireki eta itxi egingo da. Beraz, artikulazioen bidez besoari higikortasuna emango zaion bitartean, pintzarekin objektuak hartzeko ahalmena emango zaio. Diseinu arloan, kontuan eduki behar den ezaugarria besoaren higikortasuna da. Beso robotikoen kate-maila bat mugitzen denean bestea geldirik egongo da eta berdina beste kate-mailaren mugimenduarekin, beraz kate-mailak banaka mugitzen dira beraien artean guztiz independenteak izango direlarik. Beraz, hauek besoen egituren diseinuak garatzean erabili beharreko ezaugarriak izango dira.

#### *BESO TRADIZIONALAK*

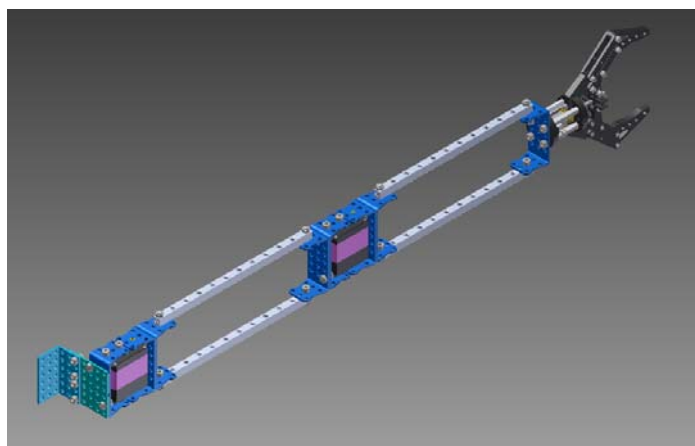
Mota honetako besoak, DC motorrak eta engranajeak erabiltzen dituzte mugitzeko. Esate baterako, DC motorraren ardatzean kokaturiko engranajea beste engranajearekin kontaktuan egon beharko da artikulazio bakoitzaren higikortasuna lortzeko. Hori dela eta, diseinatzerakoan engranaje eta motorrak besoaren beheko aldean kokatzea erabaki da. Izan ere, motorren bat artikulazio baten gainean ipiniko balitz, engranajeen arteko kontaktua modu egokian egitea oso konplexua izango litzateke, eta gainera emaitza ona ezingo litzateke ziurtatu. Horrez gain, mota hauetako besoetan engranajeak kokatzeko ardatzak beharrezkoak izango dira eta hauek ipintzeko lekua behar denez egokiena behean izatea da. Gainera mota hauetako roboten ezaugarri bereizgarria da, motor eta engranajeak besoaren beheko aldean kokatuta egotea.



*Irudia 3.5.1. Beso tradizionalaren 3D-ko diseinua*

### *TRANSMISIO ZUZENEKO BESOAK*

Beso robotiko hau eraikitzeko serbomotorrak erabiltzen dira. Serbomotoreak artikulazioetan kokatu behar dira, horretarako, serbomotorreen euskarriak erabiliko dira. Serbomotore bakoitzak bi euskarri ditu, bakoitza izkinan batean kokattuta. Alde bateko euskarria ez da mugituko, beste aldekoa, ordea, serboaren ardatzean ipiniko da eta honekin batera biratuko du. Bi artikulazio egongo direnez, bi serbomotore erabili beharko dira. Kontuan izan beharko da, beheko motorrak beso guztiko pisua eutsi beharko duela. Nahiz eta, serbomotorrek pare handiagoa izan hobe izango da behar ez den pisua ekiditea, horrela, motorraren biraketa posible izatea ziurtatuko da.

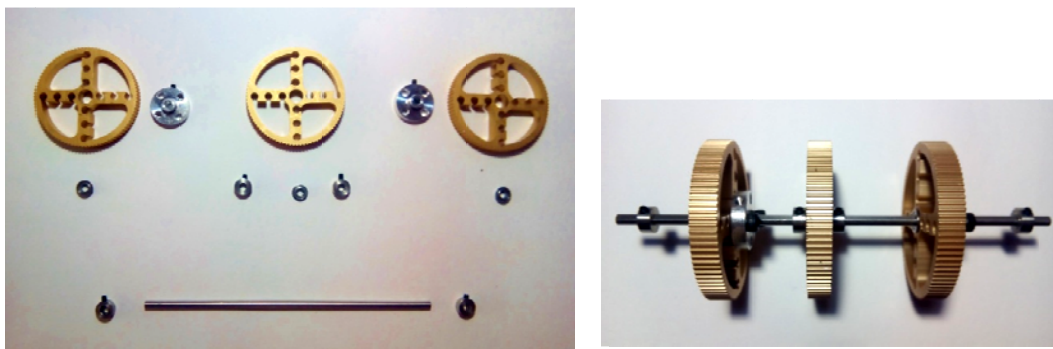


*Irudia 3.5.2. transmisio zuzeneko besoaren 3D-ko diseinua*

### 3.5.2. MUNTAKETA

#### *BESO TRADIZIONALAK*

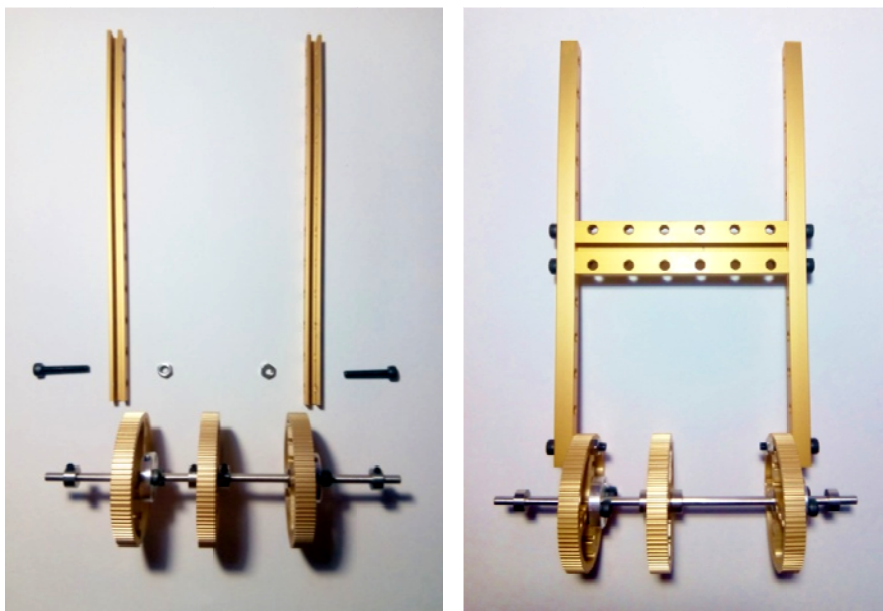
Beso tradizionalen muntaketarekin hasteko, higikortasuna emateko erabiliko diren engranajeak ardatz baten gainean jarriko dira. Horretarako behar den materiala: hiru engranaje handi, ardatz handia bat, bi ardatz konektatzaile, lau M4x14 allen torloju, lau finkapen eraztun, sei finkapen torloju eta hiru brida kojinetete. Nahiz eta, hiru engranaje ipini horietako bi bakarrik motorreko engranajearekin konektatuta egongo dira, 3. engranajea beheko kate-maila ondo eusteko izango da. Ardatzaren erdian brida kojinetetea ipiniko da, honen gainean engranaje bat kokatzen delarik. Engranajearen alde bakoitzean finkapen eraztun bat ipiniko da finkapen torlojuarekin estutuko dena. Ondoren, beste engranajeak prestatuko dira, ardatz konektatzailean finkapen torloju bat ipiniko da, baina, momentuz estutu gabe. Ardatz konektatzaile bakoitza engranajearen sartu behar da. Ardatz konektatzaileko zulo harilkatuetatik M4x14 allen torlojuak sartuko dira, horrela ardatz konektatzailea eta engranajeak baturik geratu daitezten. Pauso berdinak egin behar dira beste engranajearekin.



*Irudia 3.5.3. Ezk.ardatza eta engranajeak eraikitzeko. Esk. Ardatz eta engranajeak eraikita*

Engranaje biak prest daudenean, bakoitza ardatz nagusiko alde batetik sartzen da, erdiko engranajearekiko tarte bat uzten delarik. Hor, finkapen torlojuak estutuko da. Beharbada, aurreragoko pauso batean engranaje hauen posizioa apur bat mugitu beharko da, artikulazioaren zabalerarekin bat etor dadin. Bukatzeko, bi finkapen eraztun hartu eta finkapen torlojuarekin ardatzen izkinetan ipiniko dira. Momentuz, bi eraztunak ez dira erabiliko, baina, aurrerago ardatz nagusia euskarririk ez irteteko erabili beharko dira. Muntatutakoan, ardatzetik heldu eta erdiko engranajearen ematean honek biratu egingo du, aldiz, erdiko engranajetik eusten denean, izkinetako engranajeak ardatzarekin batera biratuko dute. Honekin ikusten da, izkinetako engranajeak ardatzera finkaturik daudela eta erdikoa, berriz, ardatzarekiko independentea dela. Beraz, engranaje bakoitzean kate-maila bat ipiniko da, gero kate-mailak beraien artean independenteak izango dira.

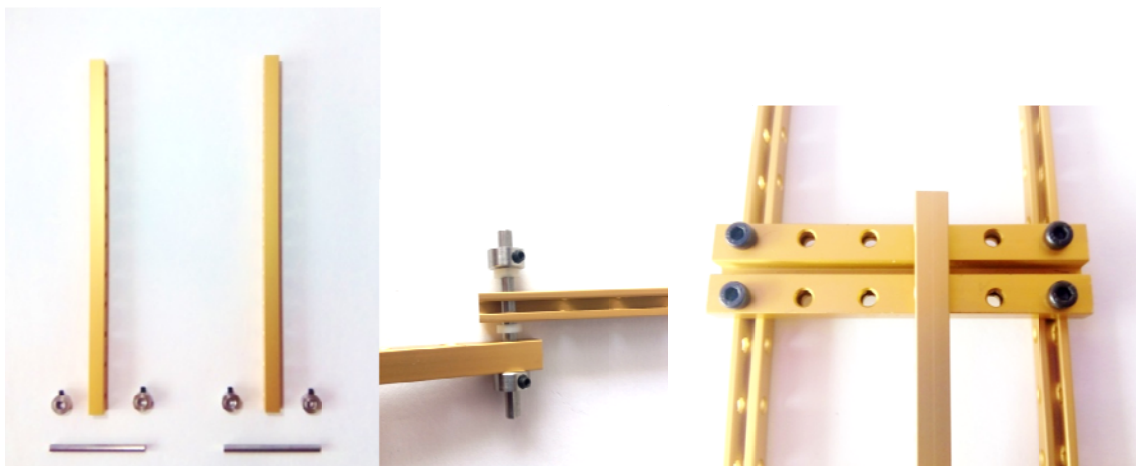
Ondoren, besoaren beheko kate-maila eraikiko da. Kate-maila hau bi izkinetako engranajeetara lotuta egongo da, horrela, engranajeak biratzean kate-mailak higiaraziko da. Kate-maila muntatzeko hurrengo materiak beharko da: 12 zuloko bi habe fina, 6 zuloko habea, bi azkoin, lau M4x14 allen torloju eta bi M4x22 allen torloju. Lehenengo pauso honetan habeak engranajeekin batuko dira. 12 zuloko habe fina hartuko da eta engranajearen kanpoko aldean kokatuko da, habearen norabidea engranajearen ardatzaren berdina izango da. Beraz, habe fineko lehenengo zuloa eta engranajearen ardatzean hortzetatik hurbilen dagoen zuloek bat egiten duten tokian M4x22 allen torloju sartuko da eta azkoin batekin finkatuko da. Beste aldekoa engranajea eta beste 12 zuloko habea erabilita prozesu bera jarraituko da. Behin habe biak engranajeetan kokaturik daudenean, biak batu beharko dira, hori egiteko 8 zuloko habea erabiliko da. Habe finen beheko zuloetatik gorantz kontaktzen hasiko da 6. eta 7. zuloa arte. Puntu horretan, habe finen artean 6 zuloko habea kokatuko da, baliteke habe finen arteko lekua nahikoa ez izatea, horrela bada, engranajetako bat mugitu beharko da. 8 zuloko habea kokatuta dagoenean 6. eta 7. zuloetan M4x14 allen torloju sartuko da. Hau egin ondoren, beheko kate-maila eraikita egongo da.



*Irudia 3.5.4. Ezk. Beheko kate-mailarako materiala Esk. Beheko kate-maila eraikita*

Beheko kate-maila egin ondoren goikoa egin behar da. Bi kate-mailen arteko mugimendua ahalbidetzeko artikulazioa ipini behar da. Artikulazioarekin bi kate-mailen arteko batzea eta mugimendua lortzen da. Guzti honetarako beharrezko materiala hurrengoa da: 12 zuloko bi habe fin, 6 zuloko habea bat, bi ardatz txiki, lau finkapen eraztun, lau

finkapen torloju, lau M4x14 allen torloju eta bi plastikozko zorro banatzailea. Artikulazioak garatzeko ardatz txikiak erabiliko dira, ardatz bat alde bakoitzean ipiniko da. Ardatza, beheko kate-maila osatzen duen habearen azkeneko zulotik eta habe horren barruko aldean, goiko kate-maila osatuko duen 12 zuloko habe fina kokatzen da, ardatza habe honen lehenengo zulotik sartzen delarik. Bi haben artean plastikozko zorro banatzailea ipini behar da, pieza guzti hauen artean distantzia txiki bat utziko da horrela arazorik gabe higitu daitezten. Ardatza irten ez dadin, alde bakoitzean distantzia batetara finkapen eraztun bana kokatuko da finkapen torloju batekin loturik. Bi aldeekin prozedura bera jarraituko da. Ondoren, habe biak batu behar dira, horretarako, 6 zuloko habea erabiliko da. Kasu honetan, bi haben artean ez da 6 zuloko habea sartuko, goiko habeek behekoekin tope egiten dutenez ez dute leku nahiko izango. 6 zuloko habea, habe finen gainean jarriko da erdi inguruan. Bertan habeko izkinetako zuloek, habe fineko errail harilkatuarekin bat egiten duten tokian M4x14 allen torlojuak sartu eta ondo estutuko dira.



*Irudia 3.5.5. Ezk. goiko kate-mailarako materiala Erd. eta Esk. Goiko kate-maila eraikita*

Orain arte beso robotikoaren bi kate-mailak eraiki dira, baina, goiko kate-maila, ardatz nagusiko erdiko engranajera konektatu behar da higitarazi dadin. Hori posible egiten duen mekanismoa eraikitzeke hurrengo materialak behar dira: 12 zuloko habe fina, 9 zuloko habe fina, habe ebakigarri bat, bost azkoin, bi M4x22 allen torloju eta hiru M4x14 allen torloju. Bi habe finak elkarrekin batuko dira, 12 zuloko habe fineko azkeneko zulotik eta 9 zuloko habe fineko lehenengo zuloek bat egiten duten puntuan M4x22 allen torloju sartuko da. Torloju hau eusteko bi azkoin ipini beharko dira, azkoin bakarra ipintzeak bi haben mugimenduak azkoina askatzea ekarriko lukeelako. Beraz, torlojuan bi azkoin ipiniko dira kontrajarrian,

horrela haben mugimenduak ez dituzte azkoinak askatuko. Hurrengo pausoarekin jarraitu baino lehen, habe ebakigarria 10. zulotik ebakiko da.

Ondoren, modu berdinean elkartuko dira 12 zuloko habe fineko lehenengo zuloa eta habe ebakigarriko azkeneko zuloa. Pauso hau egin ostean, hiru piezaz eraturiko mekanismo edukiko da, piezen arteko mugimenduak independenteak direlarik. Habe ebakigarria, erdiko engranajearekin lotu behar da. Habe ebakigarria, engranajearen ardatzaren norabide berdinean ipiniko da. Engranajearen ardatzean hortzetatik hurbilen dagoen zuloa eta habe ebakigarriko lehenengo zuloan lotu egingo dira M4X14 allen torlojua eta azkoin bat erabiliz. Orain, mekanismoaren beste aldea goiko kate-mailarekin batu behar da, horretarako, 9 zuloko habe fina 6 zuloko habearen gainean ipiniko da. Bi hauek elkartzeko, habe fineko errail harilkatua eta habeko bi zuloek bat egiten duten tokian bi M4x14 allen torloju sartu eta estutzeko dira. Baliteke aurreko pauso hau lortzeko erdiko engranajea pixka bat mugitu behar izatea.

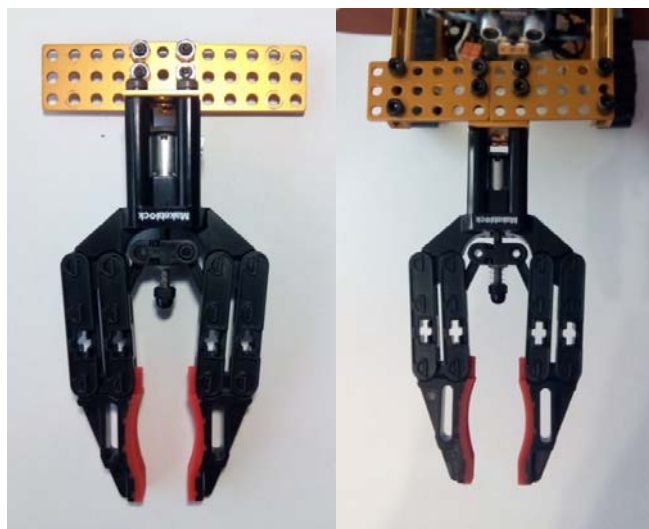
Laburbilduz, orain arte eginiko muntaketarekin besoaren kate-mailak guztiz garatu dira. Erdiko engranajea mugituz gero, goiko kate-maila gora eta behera higituko da. Beheko kate-mailak higikortasun egokia duela konprobatzeko, besoa erdiko engranajetik kontu handiarekin heldu behar da bere posiziotik mugitu ez dadin, eta izkinetako engranaje bat biratu behar da. Guzti honekin, bi kate-mailen mugimenduak independenteak direla egiaztatu da.



*Irudia 3.5.6. Beso tradizionala muntaturik*



Beso guztiaren muntaketarekin bukatzeko, goiko kate mailan pintza kokatzea besterik ez da falta. Horretarako beharrezko materiala: pintza, 3x6ko bi xafla, 3x3ko L xafla bat, lau azkoin, hamabi M4x8 allen torloju izango dira. Lehenengo, pintza besoan kokatzeko egitura eraikiko da. Horretarako, 3x6ko bi xafla ondoz ondo ipiniko dira 3 zuloako aldeetatik, hauen posizioa errespetatuz 3x3ko L xafla bi hauek elkartzen diren tokiaren gainean ipintzen dira zuloak bat egiten duten posizioan. Egitura hau batzeko lau azkoin eta lau M4x8 allen torloju erabiliko dira, hauen kokatzerakoan kontutan eduki behar da 3x3 L xaflako goiko aldean pintza kokatuko dela eta honek izkinetako lau zuloak beharko dituela. Ondorioz, 3x3 L xaflako 1, 3, 4, eta 6 zuloetatik M4x8 allen torlojuak sartuko dira eta azkoinen bidez ondo estutu. Guzti hau egindakoan egitura prest egongo da besoan kokatzeko. Lehen esan bezala, pintza, 3x3 L xaflan libre geratu den aldean kokatuko eta lau M4x8 allen torloju batuko da. Bukatzeko, eraikitako egitura goiko kate-mailaren errail harilkatuaren gainean ipiniko da. Errail harilkatua eta zuloak bat egiten duten puntuetan lau M4x8 allen torloju jarriko dira.

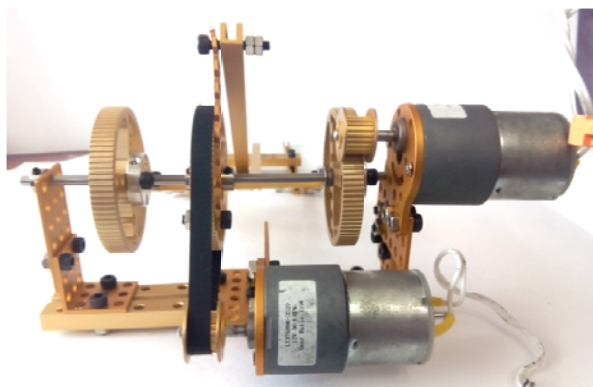


**Irudia 3.5.7.** Beso tradizionalen pintzaren kokapena

Muntaketarekin bukatzeko euskarri bat garatu behar besoarentzat eta ondoren euskarriak hori erabilia motorrak engranajeekin kontaktuan ipini behar dira. Besoaren eusleak U itxura izango du, besoaren ardatz handia kokatu ahal izateko. Euslea ez da altua izatea nahi, izan ere, honek besoa desorekatu dezake eta lurrera erortzea gauzatzea. Hori dela eta, euslearen altuera engranajeak behar duena baino zentimetro batzuk handiagoa izango da. U itxurako euslea eraikitzeko erabiliko den materiala hurrengoa da: 10 zuloako habe bat, 3x6ko bi xafla, 3x3 bi L xafla, hamasei azkoin, hamabi M4x8 allen torloju eta lau M4x14

allen torloju. Hasteko, 3x6ko xafla eta 3x3 L xafla elkartuko dira. Piezak bertikalki ipiniko dira bata bestearen gainean L xaflaren goiko zuloak 3x6ko xaflaren beheko zuloekin bat egiten duten lekuan kokatuz. Orduan, izkinetako zuloetan M4x8 allen torlojuak sartu eta azkoinenekin estutuko dira. Berdina egingo da geratzen diren beste bi xaflekin. Aurrera jarraitu baino lehen, bi pieza hauek ardatzean kokatu behar dira. Horretarako, hasiera batean ardatzean erabili gabe utzitako bi finkapen eraztunak erabiliko dira. Finkapen eraztunak askatu egingo dira, eta eraikitako piezak bakoitza barruranzko noranzkoan izkina ardatzean sartuko dira. Ondoren, pieza hauek 10 zuloko habera lotu behar dira, pieza bakoitza izkina bateko bigarren zuloetan kokatu eta bakoitzean bi M4x14 allen torloju eta bi azkoin erabiliz lotuko dira. Honen ostean, ardatzetako finkapen eraztunak ardatzean ezarriko dira finkapen torlojuak estutuz.

Besoa euskarriaren gainean ipinita dagoenean, motorrak jartzea besterik ez da geratzen. Lanean hasi baino lehen motorrak prestatu behar dira, motor bakoitzari bere euskarri ipini behar zaio eta ardatzean engranaje txiki bat jarri behar da finkapen torlojuarekin helduta mugitu ez dadin. Motorrak kokatzeko, egitura ondo behatu behar da, aukera desberdinak aztertu eta egokiena aukeratzeko. Beso hau garatzeko motorrak hurrengo toki hauetan ipini dira: beheko artikulazioa mugitzeko erabiliko den motorra, U eusleren alde batean. Baina, U eusle eta motorraren euskarriaren artean 3x6ko xafla jarri da, alde bakoitzean bi azkoin eta bi M4x8 allen torloju erabiliz. Goiko artikulazioa mugitzen duen motorra, U egituraren beheko aldean ipiniko da 10 zuloko habera erabiliz. Baina, motorraren euslea eta habearen arteko elkarketa egiteko 3x6ko L xafla erabiliko da, alde bakoitzean bi M4x8 allen torloju eta bi azkoin erabiliz batuko dira.



*Irudia 3.5.8. Besoaren euskarri eta motorrak*

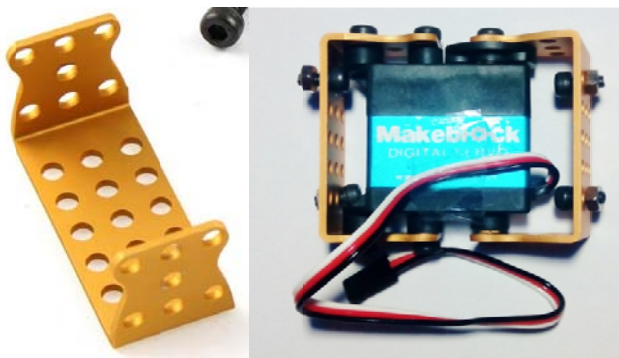
### TRANSMISIO ZUZENEKO BESOAK

Kasu honetan lehenengo pausoa serbomotoreak prestatzea da. Baina, muntaketarekin hasi baino lehen Arduino erabiliz motorrak 90 gradutara kokatuko dira. Ondoren, serbomotorearen ardatzean biribil beltza kokatuko da Irudia 3.5.6-an ikusten posizioan, indar handia egin behar da ardatza bere lekuan ondo kokatzeko.



*Irudia 3.5.9. Serbomotorearen ardatzaren kokapena*

Behin serbomotorraren ardatza prest dagoenean eusleak ipiniko dira. Euskarriak behar bezala ipintzeko materiala hurrengoa da: bi serbomotor eusle, bederatzi M4x8 Allen torlojua eta lau azkoin. Lehenengo, eusle bat hartu eta aurpegi nagusian 2. eta 14. zuloetan (eskumatik ezkerrera kontatuz) bi torlojuak sartu eta bi azkoinekin lotuko dira, ez da beharrezko asko estutzea ondorengo pauso baten erabiliko baitira. Prozesu berdina egingo da beste euslearekin, biak prest daudenean serbomotorrean ipiniko dira. Lehendabizi motorreko eusle finkoa kokatuko da, hau da, motorra ez dagoen ardatzaren aldetik sartu behar da. Euskarriko zuloak, motorreko goiko aldean eta behekoan dauden bi zuloekin bat egiten duten toki bakoitzean M4x8 Allen torlojua jarriko dira. Torloju guztiak estutu ondoren euskarria motorrera finkaturik egongo da. Beste euskarria kokatzeko ardatz biribileko bi irtenguneak eusleko lehenengo bi zuloetan sartu behar dira, ondoren, behe aldeko irtengunea kokatu behar da. Goiko aldean, ardatz biribileko erdian eta eusleko zuloak batzeko M4x8 Allen torlojua ipiniko da. Honekin bigarren euslea jarrita egongo da eta biratzen duela egiaztatu daiteke. Beste serbomotorrari eusleak modu berdinean ipiniko dira.



*Irudia 3.5.10. Ezk. Serboaren euskarria Esk. Serbomotorra euskarrienkin*

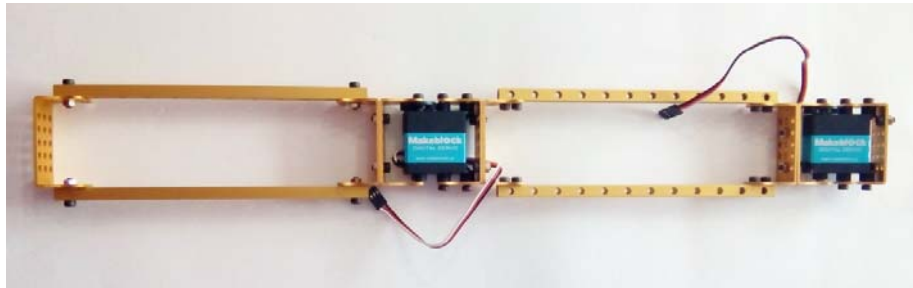
Serbomotoreei eusleak ipiniz artikulazioak garatu dira, hurrengo pausoa kate-mailak eraikitzea da. Hauek garatzeko beharrezko materiala honako hau da: lau serbomotore eusle, lau 12 zuloko habe fin, zortzi M4x14 Allen torlojua eta zortzi azkoin. Bi kate-mailak egingo dira, beraz, ondorengo pausoak bi aldiz egin beharko dira. Motor euslea eta bi 12 zuloko habe fin dira. Euslearen izkinetako erdiko zuloa eta habeko lehenengo zuloa bat egiten duten puntuan M4x14 Allen torlojua sartuko da eta azkoin batekin lotuko da. Berdina egingo da euslearen beste aldean eta baita beste habearekin. Besoa eraikitzen bukatzeko habeen azkeneko zuloak beste motor euslearekin modu berdinean batu beharko dira.



*Irudia 3.5.11. Transmisio zuzeneko kate-maila*

Orain arte, transmisio zuzeneko besoa osatzen duten bi kate-maila eta artikulazioak eraikita dira. Beraz, ondoko pauso honetan biak batuko dira, horretarako, hasieran motorreko euskarrietan utzitako torloju eta azkoinak erabiliko dira. Beheko artikulaziotik hasiko da, hau hartu eta eusle mugikorreko torlojuetako azkoinak askatuko dira. Kate-maila bat hartu eta kontu handiarekin bertako euslea eta motorrarena bat egingo dira. Artikulazioko torlojuak, kate-mailaren eusletik zeharkatuko dute eta azkoinekin bi zatiak batuko dira. Kate-mailaren beste aldean beste artikulazio jarriko da modu berdinean, baina oraingoan serbomotorrean

mugitzen ez den euslearen aldetik egingo da. Beso guztiaren muntaketarekin bukatzeko, azken artikulazioko eusle mugikorra eta goiko kate-maila elkartuko dira.



*Irudia 3.5.12. Transmisio zuzeneko besoa muntaturik*

Transmisio zuzeneko beso robotikoaren eraikuntzarekin bukatzeko pintza kokatzean falta da. Horretarako beharrezko materiala pintza eta lau M4x8 allen torloju izango dira. Pintza, goiko kate mailaren goiko euskarrian kokatuko da, euskarriko eta pintzaren zuloek bat egiten duten tokian M4x8 allen torlojuak sartuko dira



*Irudia 3.5.13. Transmisio zuzeneko besoan pintzaren kokapena*

Azkenik muntaketarekin bukatzeko besoa finkatzeko oinarri bat beharko da. L itxurako egitura eraikitzeko, beharrezko materiala hurrengoa da: 3x6ko L xafla bat, 3x6ko bi xafla, lau M4x8 Allen torlojuak eta lau azkoin. L egitura eraikitzeko, 3x6ko L xafla eta 3x6ko bi xafla hartu, azken hauek bata bestearen ondoan gorantz begira ipiniko dira. Ondoren bi xafla hauek 3x6ko L xafla gainean ipini, L xaflako lehenengo bi lerroak beste xaflako azkeneko lerroekin bat egiten duten puntuak lau torlojuak sartuko dira eta lau azkoinekin estutuko dira. Egitura egin ostean, besoa jarriko da. Beheko artikulazioan serbomotorrean

mugitzen ez den euslea aske geratu da, beraz, bertako azkoinak askatuko dira. Motorraren euslea L egituraren goiko aldean kokatuko da eta azkoinak estutuko dira.

### 3.5.3. PROGRAMAZIOA

Programazioari ekin baino lehen esan beharra dago bi besoen programak helburu bera izango dutela. Programen helburua, hain zuzen, besoaren higikortasuna aginte infragorritz kontrolatzea izango da. Programen arteko desberdintasunak motor mota desberdinak erabiltzen direnez, hauek kontrolatzeko komando desberdinak erabili behar izango direla da. Beso tradizionalen DC motorrak erabiltzen diren bitartean, transmisio zuzenekoetan serbomotorrak erabiltzen dira.

Besoen programazioan zehar, une oro, robotak guztiz muntaturik dagoenean bukaerako programa nagusia kontutan edukiko behar da. Honek suposatzen du, oinarrian, plakako konexioak egiteko erabiliak izan diren pinak eta aginte infragorriko botoi berdinek erabili beharrean beste batzuk erabiliko direla amaierako muntatzea errazteko. Bestela, bukaerako programa nagusian portu eta aginteko infragorriko botoiak aldatzen hasiz gero nahaste handia sortuko litzateke. Aginte infragorriko beste botoi batzuk erabiltzeak programazioan beste botoiak izendatzearekin nahiko izango da. Baina, plakako konexioak aldatzerakoan, programazioan beste portuak izendatzeaz gain, konexioak aldatu beharko dira. Kasu hauetan, DC motorrak zein serbomotorrak 1. eta 2. pinetara konektatu behar dira, baina oraingoan, motorrak ezin dira zuzenean portuetara konektatu tartean DC motorren edo serbomotorren driverra ipini behar delako. DC motorren driverrak motor bakarra konektatzeko aukera ematen du. Baina serbomotorren driverrek, ordea, bi serbomotore konektatzeko aukera eskaintzen dute, gainera, beraien artean independenteak dira, hau da, bakoitzak biraketa desberdinak egin ditzake. Beraz, besoak garatzerakoan, motorrak 1. eta 2. pinetara konektatu behar direnez, 2 driver beharrezkoak izango dira. Hori dela eta, hurrengo modu honetan egingo dira konexioak: RJ25 kablearekin 1. portua eta driver baten sarrerarekin konektatuko da ondoren, goiko artikulazioa mugiarazten duen motorrak sarreran daukan kablea driver-raren irteera konektatuko da. Konexio berdinek egin behar dira beste driver-rarekin, baina, 2. portua eta beheko artikulazioa higiaraziko duen motorra erabilia.

Serbomotorrak erabiltzen direnean, driver bakoitzak bi konexio egiteko aukera duenez driver bakarra erabilia nahikoa izango da.

Baina, konexio guzti hauek egindakoan, besoen bukaeran kokaturiko pintzarena egitea faltako da. Pintza honek objektuak maneiatzeko aukera eskaintzen du. Pintza ireki eta ixteko pintzara akoplaturiko DC motor txiki bat aurkitzen da. Beraz, motorra alde batera biratzean pintza irekiko da eta beste aldera biratzen duenean pintza itxi egingo da. Pintzaren espezifikazioen arabera, DC motor honek 5V eta 12V bitarteko tentsioarekin funtzionatzen du. Hori dela eta, plakako 1 edo 2 pinera konektatu beharko lirateke. Transmisio zuzeneko besoetan ez da arazorik egongo, izan ere, serbomotoreak 1 portura konektatu dira driver batean. Beraz, pintzako DC motorra DC driver bat erabiliz 2. pinera konektatzea posible izango da. Arazoa beso tradizionala garatzerakoan edukiko da 1. Eta 2 pinak okupatuta egongo direlako besoko kate-maila mugiaraziko dituzten DC motorrekin. Arazo honen konponbidea beso tradizionalaren atalean emango zaio.

Era berean, programazioa argiagoa eta ulergarriagoa izan dadin, oinarrian kokaturiko motorrei *m\_esk* eta *m\_ezk* izena jarri zitzaenez, besokoei izen desberdina ipiniko zaie DC motorretan *m\_goi* eta *m\_behe* eta serbomotorretan *serbo\_goi* eta *serbo\_behe*. Horrela motor bakoitzak goiko edo beheko artikulazioa higitzeko erabiltzen den jakin ahal izango da.

## BESO TRADIZIONALAK

Programa nagusiarekin hasi baino lehen, osagai elektroniko guztiek, driver eta motorrek, funtzionatzen dutela egiaztatu behar da. Honetarako proposena, motorrak besoko egiturari ipini baino lehen egitea da, bestela, motorrek denbora luzeegian biratzen badute besoa hondatu dezakete, besoren biraketa mugatuta dagoelako. Motorrak eta kasu honetan, hauek 1 eta 2 portuan konektatuko direnez motorren driverrak ere, egiaztatzeko erabiliko den programa *beso\_egiazt* izango da. *beso\_egiazt* programarekin lehenengo motorrak alde batera biratuko dute 1 segundoz; gero, geldirik egongo da 3 segundoz; ondoren, 1 segundoz beste aldera biratzeko eta azkenik 6 segundoz geldirik egongo dira programa berriz exekutatzeko hasi arte.

```

/*****
Programaren izena: beso_egiazt
Egilea: Itziar Aldekoa
Deskribapena: Programa honekin erabiliko diren motorrak eta driverr
funtzionatzea dutela ziurtatuko da.

*****/
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeDCMotor m_goi(PORT_1); //1.pinean goiko artikulazioa mugitzen due
MeDCMotor m_behe(PORT_2); //2.pinean beheko artikulazioa mugitzen du

byte motor_abid = 190; // motorren abiadura, PWM-a balio maximoa

void setup()
{

}

void loop()
{
    m_goi.run(motor_abid); // motorrak alde batera biraka
    m_behe.run(motor_abid);
    delay(3000); // 3 segunduz
    m_goi.stop(); //motorrak geldirik
    m_behe.stop();
    delay(1000); // 1 segunduz
    m_goi.run(-motor_abid); //motorrak beste aldera biraka bir
    m_behe.run(-motor_abid);
    delay(3000); // 3 segunduz
    m_goi.stop(); //motorrak geldirik
    m_behe.stop();
    delay(6000); // 6 segunduz
}

```

*Irudia 3.5.14. beso\_egiazt programa*

Programa exekutatzean eduki daitekeen arazorik larriena motorrek ez biratzea da. Hau arrazoi desberdinengatik izan daiteke: pilak Arduino plakara konektatu direla MAKEBLOCK-eko plakara konektatu beharrean, edo pilak agorturik egotea, edo konturatu gabe programa plakan ez konektatzea, edo etengailua ON posizioan ez egotea. Exekutatzean izan daitekeen beste arazoetariko bat motor batek ez biratzea izan daiteke, hau gertatuz gero, driver-ren argiak begiratu behar da, DRC argia pizturik badago eta PWM argia itzalita, honek adierazten du RJ25 kablea ez dela ondo konektatu, beraz, kablea atera eta berriz ondo konektatu beharko da.



Laburbilduz, *beso\_egiazt* programarekin bukaerako programa nagusia kontuan izanik, driver-ren konexio eta erabilera zuzena egiaztatu da. Horrez gain, driverren eta DC motorren funtzionamendua ere zuzena dela ziurtatu da.

Ondorengo programaren helburua, aginte infragorrien bidez besoa mugiarazten duten DC motorrak kontrolatzea izango da. Modulu infragorria oinarrian bezala 3.pinean konektatuko da. Hasieran esan bezala, aginte infragorriko beste botoi batzuk erabili beharko dira, horregatik zenbakidun botoiak erabiliko dira. Aginteko 1 eta 4 botoiak erabilia goiko kate-maila kontrolatuko da, 1 botoiari ematean goiko kate-maila gorantz mugituko da eta 4 botoiari ematean beherantz mugituko da. Beheko kate-maila kontrolatzeko, aldiz, 2 eta 5 botoiak erabiliko dira, 2 botoiari ematean beheko kate-maila gorantz mugituko da eta 5 botoiari ematean beherantz mugituko da. Objektuak maneiatzeko pintza 3 botoiarekin ireki egingo da eta 6 botoiarekin itxiko da. Guzti hauek kontuan izanda *beso\_infrag* programa garatu da.

Baina programa garatu baino lehen besoarekin bukaeran kokatuko den pintzaren konexioak egin beharko dira eta hauek ondo funtzionatzen dutela ziurtatu beharko da. Atal honen hasieran aipatu bezala, beso tradizionalen kasuan pintzaren konexioa egiterakoan arazoa edukiko da. Izan ere, pintza konektatzeko pinak 1. edo 2. izan beharko lirateke, baina, pin hauetan besoko DC motorrak aurkitzen dira. Arazo honi konponbidea emateko RJ modulua erabiltzea pentsatu da, modulu hauen irteeran 5V daudelako. RJ moduluen irteeran hiru pineko 2 konexio daude, irteera pin hauetan GND,  $V_C$  (+5V) eta S seinalea aurkitzen dira. Pin mota hauek, beste arazo bat gehitzen dute, izan ere, DC motorrak bi konexio bakarrik dituztelako positiboa eta negatiboa edo, kasu honetan. 5V eta GND (0V) izango direnak. Motorrak alde batera eta bestera biratzeko konexio hauek programazioaren bidez aldatzeko aukera izan behar dute. Baina, RJ moduluetako GND eta  $V_C$  finkoak dira, ez dago programazioaren bidez elkar trukatzeko aukerarik. Ondorioz, RJ moduluko S1 eta S2 seinaleak erabiltzea erabaki da, hauek programatzeko aukera eskaintzen dutelako. Beraz, kable baten bidez 6 pinean RJ modulua konektatu da eta pintzako DC motorreko konexioak moduluko S1 eta S2-ra egin dira. Konexio hauekin pintzaren funtzionamendua egokia dela ziurtatzeko *pintza\_egiazt* programa garatu da.

```

/*****
Programaren izena: pintza_egiazt
Egilea: Itziar Aldekoa
Deskribapena: Programa honen helburua, beso tradizionalen pintzare
funtzionamendua zuzena egiaztatzea da.
*****/

#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MePort  pintza(PORT_6);

void setup()
{

}

void loop()
{
  pintza.Dwrite1(HIGH);
  pintza.Dwrite2(LOW);
  delay(3000);
  pintza.Dwrite1(LOW);
  pintza.Dwrite2(HIGH);
  delay(3000);
}

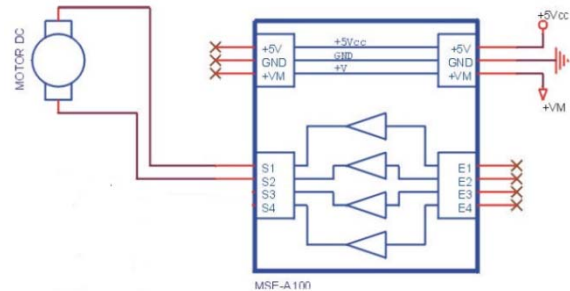
```

*Irudia 3.5.15. pintza\_egiazt programa*

*Pintza\_egiazt* programan pintzaren funtzionamendua ziurtatzerakoan, pintza irekitzen eta ixten hasten da, baina, oso geldo mugiarazten da eta badirudi ez daukala indar handirik. Hau da, pintza irekitzen edo ixten hasten da eta handik gutxira indarra galtzen doa gelditu egiten delarik. Horrez gain, batzuetan motorra ez da gai biratzen hasteko. Hau 5Veko tentsioak emandako korrontetik pintzara beharrezkoa den kantitatea ez delako iristen gertatzen da, hori dela eta, motorrak ez du modu egokian biratzeko indar nahikorik izango. Arazo honen konponbidea, RJ modulua eta pintzako DC motorraren artean driver bat kokatzea da. Erabiliko den driverra MSE-A100 izango da. Driver honetan egingo diren konexioak hurrengoak izango dira. RJ moduluko S1 eta S2 pinak driver-reko E1 eta E2 pinetara konektatuko dira, eta driver-reko S1 eta S2 irteerak pintzaren DC motorreko sarrerara. Beste alde batetik, driverra elikatu egin beharko da horretarako 5V eta GND-ra RJ modulora konektatuko dira. Eta azkeneko konexioa dirver-reko  $V_M$  plakako elikadurara konektatu beharko da, tentsio hau driver-reko S1 eta S2 irteeren artean edukiko dena izango

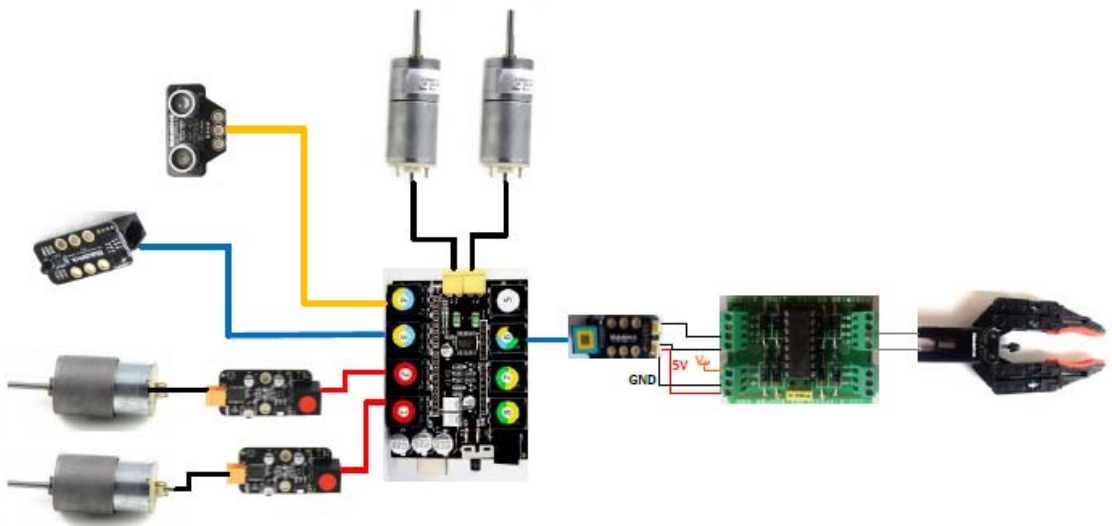
da. Driver-rearen honela funtzionatzen du: E1 eta E2 sartzen diren seinaleak berdinak direnean motorrak ez du biratuko. E2 aktibo dagoenean eta E1 ez, ordea, erlojuen ardatzen norabide berdinean biratuko du. E1 aktibo dagoenean eta E2 ez, motorrak erlojuen ardatzeren aurkako norabide berean biratuko du. Konexio guzti hauek egin ondoren, berriz ere, *pintza\_egiazt* programa erabiliko da pintzaren funtzionamendu egokia ziurtatzeko. Oraingo honetan ikusiko da nola pintza indar gehiagorekin biratzen duen eta ez den bidean geratuko.

E1	E2	MOTORRA
0	0	OFF
0	1	Biraketa erlojuko ardatzen norabidean
1	0	Biraketa erlojuko ardatzen aurkako norabidean
1	1	OFF



*Irudia 3.5.16. Driverreko konexioak*

Beraz, pintzaren konexio guztiak egin ondoren eta funtzionatzen duela ziurtaturik dagoenean, plakan hurrengo konexio hauek daudela ziurtatu beharko da.



*Irudia 3.5.17. Beso tradizionaleko konexioak*

Konexio guztiak ondo eginik daudela ziurtatzean lehen azalduko *beso\_infrag* programa plakan kargatuko da, bere funtzionamendua zuzena ziurtatzeko

```

/*****
Programaren izena: beso_infrag
Egilea: Itziar Aldekoa
Deskribapena: Beso tradizionala aginte infragorri bidez kontrolatu
Horretarako, erabiliko diren botoiak 1,5,2,4,3 eta 6 izango dira di
*****/
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeDCMotor m_goi(PORT_1); //1.pinean goiko kate-maila mugitzeko mot
MeDCMotor m_behe(PORT_2); //2.pinean beheko kate-maila mugitzeko mot
MeInfraredReceiverinfragorri_hartz(PORT_3);
//hartzaile infragorria 3.pinera konektatzen da.
MePort pintza(PORT_6); // Pintzake RJ modulura konektaturik

byte motor_abid = 200; // motorren abiadura, PWM-a balio maximoa

void setup()
{
  infragorri_hartz.begin();
}

void loop()
{
  if(infragorri_hartz.available() || infragorri_hartz.buttonState())
  {
    switch(infragorri_hartz.read())
    {
      case IR_BUTTON_1:// 1 botoia eman, goiko kate-maila goran
        Gorantz1();
        break;
      case IR_BUTTON_4:// 4 botoia eman, goiko kate-maila beheara
        Beherantz1();
        break;
      case IR_BUTTON_2:// 2 botoia eman, beheko kate-maila goran

```

*Irudia 3.5.18. beso\_infrag programa (1.zatia)*

```
//Goian izendatutako botoiari sakatzean exekutatuko den programak
void Gorantz1()// Goiko kate-maila gorantz mugitu horretarako
{
    m_goi.run(motor_abid); // goiko motorra biratu
    m_behe.stop(); // beheko motorra geldik
}

void Beherantz1()// Goiko kate-maila beheantz mugitu horretarako
{
    m_goi.run(-motor_abid); // goiko motorra beste aldera biratu
    m_behe.stop(); // beheko motorra geldik
}

void Gorantz2()// Beheko kate-maila gorantz mugitu horretarako
{
    m_goi.stop(); // goiko motorra geldik
    m_behe.run(-motor_abid); // beheko motorra beste aldera biratu
}

void Beherantz2()// Beheko kate-maila beherantz mugitu horretarako
{
    m_goi.stop(); // goiko motorra geldik
    m_behe.run(motor_abid); // beheko motorra biratu
}

void pintza_ireki()// Pintza irekiko da horretarako
{
    pintza.Dwrite1(HIGH); // S1 seinalea aktibatuko da
    pintza.Dwrite2(LOW); // S2 seinalea ez da aktibatuko
}

void pintza_itxi()// Pintza itxiko da horretarako
{
    pintza.Dwrite1(LOW); //S1 seinalea ez da aktibatuko
    pintza.Dwrite2(HIGH); //S2 seinalea aktibatuko da
}

void Stop()// besoa gelditzeko motore guztiak geldik
{
    m_goi.stop();
    m_behe.stop();
    pintza.Dwrite1(LOW);
    pintza.Dwrite2(LOW);
}
```

*Irudia 3.5.19. beso\_infrag programa (2.zatia)*

Laburbilduz, programa honekin besoaren higikortasuna aginte infragorri baten bidez kontrolatzea lortu da. Programa besoan egiaztatzekoan ikusi da, egitura egonkorra dela eta besoaren mugimenduak egokiak direla ez dutelako oszilatzen. Diseinu atalean esan den bezala, kate-mailak beraien artean independenteki mugitzen direla ziurtatu da, hau da, beheko kate-maila mugitzean goikoa ez dela mugitzen eta alderantziz. Horrela, diseinu atalean beso robotikoarentzat ezarritako eginkizun guztiak bete direla egiaztatu da. Gainera, besoa luzatu eta batu daiteke inolako arazorik gabe. Hala ere, kontuan eduki beharko dira besoa guzti luzaturik edo baturik dagoenean botoiari gehiago ez ematea hondatu ez dadin, beste aldera eman ahal zaio baina ez alde berdinerantz. Bestalde, pintzarekin izandako arazo guztiak konpontzea lortu da RJ modulu eta pintzako DC motorren artean kanpoko driver bat ipiniz. Orain ez duela arazorik ematen eta beso guztiarekin bat egiten duela egiaztatu da.

## TRANSMISIO ZUZENeko ROBOTAK

Transmisio zuzeneko robotaren programazioa egiteko serbomotoiak erabiliko dira, orain arte erabili ez den elementua da. Hori dela eta, programa nagusiarekin hasi baino lehen oinarritzko programak garatuko dira. Programa honek bi helburu izango dituzte, batetik, serbomotorrak ondo funtzionatzen dutela ziurtatzea eta bestetik, hauek programatzeko komandoen erabilera egokia aztertzea da. Serbomotorren eta DC motorrak programatzeko orduan zenbait aldaketa daudela ikusiko da, esate baterako, modu desberdinetan izendatzen dira, pin konexioak ezartzerakoan serbomotorren driverrak bi motor konektatzeko aukera eskaintzen duela edo motorrak biratzeko komandoa desberdinak dira. Gauza guzti hauek behatzeko hobe da serbomotorrak besora konektaturik ez egotea. Beste serbomotore batzuk erabiliko dira hobeto aztertzeko aukera edukitzeko.

```

/*****
Programaren izen: serbol
Egilea_Itziar Aldekoa
Deskribapena: Serbomotorra programatzen ikasiko da, horretarako, serboen
komando desberdinak erabiliko dira eta hauel aztertu.
*****/

#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>
//Serbomotorraren driverra 1. portura konektatu
MeServo serbo_goi(PORT_1,DEV1);
// Driverreko 1.pinean goiko artikulazioa mugitzeko motorra
MeServo serbo_behe(PORT_1,DEV2);
// Driverreko 2.pinean beheko artikulazioa mugitzeko motorra

byte pos1 = 0;//1. posizioa 0 gradutan
byte pos2 = 180;//2. posizioa 180 gradutan

void setup()
{
  serbo_goi.begin();
  serbo_behe.begin();
}

void loop()
{
  // 1.PROGRAMA
  serbo_goi.write(pos1);// goiko artikulazioa pos1-era biratu
  serbo_behe.write(pos1);// beheko artikulazioa pos1-era biratu
  delay(1000);
  serbo_goi.write(pos2);// goiko artikulazioa pos2-era biratu
  serbo_behe.write(pos2);// beheko artikulazioa pos2-era biratu
  delay(1000);

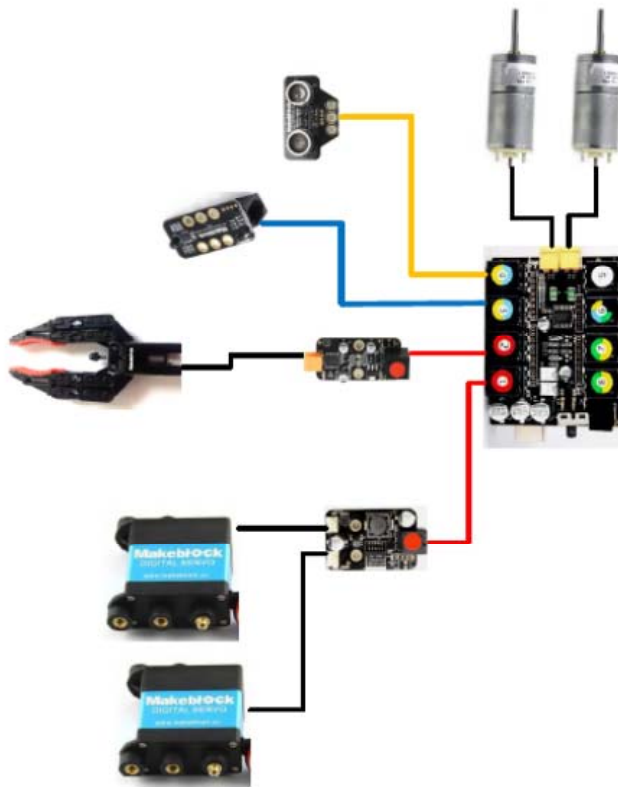
  //2.PROGRAMA
  serbo_goi.writeMicroseconds(1000);
  serbo_behe.writeMicroseconds(1400);
  delay(1000);
  serbo_goi.writeMicroseconds(2000);
  serbo_behe.writeMicroseconds(1600);
  delay(1000);
}

```

*Irudia 3.5.20. serbol programa*

*Serbol* programaren barruan, bi azpiprograma daudela ikusi daiteke. Lehenengo, *1.programa* ipintzen duen atala exekutatu da eta *2.programa* iruzkin bezala ipini da eta gero alderantziz. Hasierako atala *void loop()*-era iritsi arte bietan berdina izango da, beraz, hau aztertzen hasiko da. Aurrerago esan bezala, oraingo honetan serbomotorea izendatzeaz eta

konexioa ipintzeaz gain, driverreko zein pinetara konektatu den adierazi behar da, horretarako, *DEV\_1* eta *DEV\_2* erabiltzen dira. Ondoren, void *setup()*-en barruan driverrak hasieratzen dira. Azkenik, bi programak aurkitzen dira, kasu bietan bi motorrei biratzeko agindua eman ostean delay dagoela ikusten da, modu honetan biek batera biratzen dute. Aldiz, bi aginduen erdian beste delay bat ipiniz gero, lehenengo goiko artikulazioa biratuko du eta delay-an ezarritako denbora pasatu ostean, bigarrenak biratuko du. Azkenik, delay guztiak kenduz gero instrukzioak arinegi exekutatzen dira serbomotorrak birarazi ahal izateko. Bi programen arteko desberdintasuna aginduetan aurkitzen da batean *.write* eta bestean *.writeMicroseconds* da. Lehenengo kasuan, parentesien arteko balioa graduak dira 0 eta 180 bitartekoa da eta bigarren kasuan, aldiz, 1000 eta 2000 artekoa. Azkeneko balio horiek 1000 eta 2000 mikrosegundoak dira eta pultsuaren balio altuaren iraupena adierazten du. Guzti honekin, serboen programazio aukera desberdinak aztertu dira eta ondo funtzionatzen dutela egiaztatuz.



*Irudia 3.5.21. Transmisio zuzeneko besoaren konexioak*



Garatuko den hurrengo programan osagai elektronikoko guztiak erabiliko dira. Hori dela eta, garrantzitsua izango da guztia ondo konektaturik dagoela ziurtatzea, beraz, hasieran azaldutako konexioez gain pintza ere konektatu beharko da. Atalaren hasieran esan bezala, transmisio zuzeneko besoa garatzean plakan pintzaren konexioak egiteko ez da arazorik edukiko, izan ere, plakako 2. pineko konexioa libre dagoelako. Beraz, 2. pin honetatik kable batekin DC driverraren sarrerara konektatuko da eta pintzako DC motorra driver honen irteerara konektatuko da. Ondorengo irudian (ikus irudia 3.5.19) konexio guztiak adierazten dira.

Landutako serbomotoreen programazioa transmisio zuzeneko besoaren programazioa garatzeko aplikatuko da. *Besoserb\_infrag\_2* programaren helburua, aginte infragorri baten bidez beso robotikoa mugiaraztea da. Transmisio zuzeneko besoaren artikulazioetan serbomotoreak daude eta hauek dira besoa eusten dutenak, ez dago beste inongo egiturarik. Beraz, motorrak martxan ez daudenean besoak ez du indarrik zutik egoteko. Hori dela eta, programaren hasieran besoaren, hasierako posizioa bat ezarriko zaio. Horretarako, besoen ohiko posizio bat garatzea erabaki da: beheko motorra 120 gradutara eta goiko motorra 40 gradutara kokatuko dira. Eta ondoren, aginte infragorritzko kontrola garatuko da. Aginte infragorria, aurreko kasu berdinean bezala kontrolatuko 1 eta 4 botoiek goiko kate maila kontrolatuko dute 1 botoari ematean gorantz higituko da eta 4 botoiari ematean beherantz. Beheko kate-maila kontrolatzeko 2 eta 5 botoiak erabiliko dira, 2 botoiari ematean gorantz eta 5 botoari sakatzean beherantz higitzen delarik. Botoi bakoitzari ematean motorrak angelu konkretu bat biratu beharko du, bestea, bere posizioan geldirik mantentzen den bitartean. Kontuan izan behar da, biratuko duen angelua txikia izango dela, angelua txikia bada ere barra hauen luzera handia da 184 mm-takoa, beraz, habeak gehiago biratzen du. Horrez gain, angeluaren balioa txikia izan behar du doitasun egoki bat lortzeko, proba desberdinak egin ondoren 2 gradutako angelua egokia dela erabaki da. Bestalde, 3 eta 6 botoiak besoaren bukaeran kokaturiko pintza kontrolatzeko erabiliko dira. 3 botoiari ematean pintza irekiko da eta 6ari ematean, ordea, itxi egingo da. Pintzako DC motorra programatzeko beste edozein DC motor bezala programatuko da.

```

/*****
Programaren izena: besoserb_infrag
Egilea: Itziar Aldekoa
Deskribapena: Transmisio zuzeneko besoa aginte infragorri bidez ko
da. Horretarako, erabiliko diren botoiak 1,5,2,4,3 eta 6 izango dira
*****/

#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

//Serbomotorraren driverra 1. portura konektatu
MeServo serbo_goi(PORT_1,DEV1);
// Driverreko 1.pinean goiko artikulazioa mugitzeko motorra
MeServo serbo_behe(PORT_1,DEV2);
// Driverreko 2.pinean beheko artikulazioa mugitzeko motorra
MeInfraredReceiverinfraredReceiverDecode(PORT_3);
//hartzaille infragorria 3.pinera konektatzen da.
MeDCMotor pintza(PORT_2);// Pintzako DC motorra

byte pos1 = 40;// Goiko artikulazioan 40 gradutara ipini
byte pos2 = 120;// Goiko artikulazioan 120 gradutara ipini
byte angl;
byte ang2;

void setup()
{
  serbo_goi.begin();
  serbo_behe.begin();
  infraredReceiverDecode.begin();
  // Besoaren hasierako posizioa ezarri
  serbo_goi.write(pos1);
  serbo_behe.write(pos2);
}

void loop()
{
  pos1 = serbo_goi.read();

  pos2 = serbo_behe.read();

if(infraredReceiverDecode.available() || infraredReceiverDecode.buttc
{
  switch(infraredReceiverDecode.read())
  {
    case IR_BUTTON_1:// 1 botoia eman, goiko kate-maila gorantz m
      Gorantz1();
      break;
    case IR_BUTTON_4:// 4 botoia eman, goiko kate-maila beherantz :
      Beherantz1();
      break;
    case IR_BUTTON_2:// 2 botoia eman, beheko kate-maila gorantz m
      Gorantz2();
      break;

```

*Irudia 3.5.22. besoserb\_infrag programa(1.zatia)*

```

    case IR_BUTTON_5:// 5 botoia eman, beheko kate-maila gorantz mu
        Beherantz2();
        break;
    case IR_BUTTON_3:// 3 botoia eman, pintza irekiko da
        pintza_ireki();
        break;
    case IR_BUTTON_6:// 6 botoia eman, pintza itxiko da
        pintza_itxi();
        break;
    }
}
else
{
    Stop();
}
}

//Goian izendatutako botoiari sakatzean exekutatu den programak
void Gorantz1()// Goiko kate-maila gorantz mugitu horretarako
{
    if(pos1 < 185)//goiko artikulazioaren angelua 185 baino txikiagoa
    {
        angl=pos1;
        angl=angl+2;// 2 gradu gehitu angeluari
        serbo_goi.write(angl);//kalkulaturiko angelura mugitu goiko serb
        serbo_behe.write(pos2);// beheko serbo zeukan angeluan mantendu
        delay(5);
    }
}

void Beherantz1()// Goiko kate-maila beheantz mugitu horretarako
{
    if(pos1 > 0)//goiko artikulazioaren angelua 0 baino handiagoa den
    {
        angl=pos1;
        angl=angl-2;// 2 gradu kendu angeluari
        serbo_goi.write(angl); //kalkulaturiko angelura mugitu goiko ser
        serbo_behe.write(pos2);// beheko serbo zeukan angeluan mantendu
        delay(5);
    }
}

void Gorantz2()// Beheko kate-maila gorantz mugitu horretarako
{
    if(pos2 < 185)//beheko artikulazioaren angelua 185 baino txikiago
    {
        ang2=pos2;
        ang2=ang2+2;// 2 gradu gehitu angeluari
        serbo_goi.write(pos1);// goiko serbo zeukan angeluan mantendu
        serbo_behe.write(ang2);//kalkulaturiko angelura mugitu beheko se
        delay(5);
    }
}
}

```

*Irudia 3.5.23. besoserb\_infrag programa(2.zatia)*

```

void Beherantz2()// Beheko kate-maila beherantz mugitu horretarako
{
  if(pos2 > 0)//goiko artikulazioaren angelua 0 baino handiagoa den
  {
    ang2=pos2;
    ang2=ang2-2;// 2 gradu kendu angeluari

    serbo_goi.write(pos1);// goiko serbo zeukan angeluan mantendu
    serbo_behe.write(ang2);//kalkulaturiko angelura mugitu beheko se
    delay(5);
  }
}
void pintza_ireki()// Pintza irekiko da horretarako
{
  pintza.run(100); // motorrak alde batera biratu
}

void pintza_itxi()// Pintza itxiko da horretarako
{
  pintza.run(100); //motorrak beste aldera biratu
}

void Stop()// besoa gelditzeko motore guztiak geldik
{
  serbo_goi.write(pos1);// goiko serbo zeukan angeluan mantendu
  serbo_behe.write(pos2); // beheko serbo zeukan angeluan mantendu
  pintza.stop();//pintza geldik mantentzeko
}

```

*Irudia 3.5.24. besoserb\_infrag programa(3.zatia)*

*Besoserb\_infrag\_2* programa beso robotikoan egiaztatzerakoan behatu da: serbomotorrek pare handia dutenez, besoak indar handia duela. Indar handia izateak pisu gehiago mugi dezakeela suposatzen du, hala ere, kontutan izan behar da beso mota hauetan pisua eusteko eta orekaturik mantentzeko serbomotorra besterik ez daudela. Baina, aldi berean indar guzti horrek besoa desorekatzea ekar dezake. Gainera, mugimendu bakoitzean biratutako angelua txikia denez, kolpeka mugitzen dela dirudi. Bestalde, serbomotorrek beraien posizioa mantentzerakoan dar-dar pixka bat egiten dute, honek besoak ere dar-dar egitea dakar, hau da, oszilazio ez desiragarriak ditu posizioaren inguruan. Guzti honez gain, inertziaren ondorioz besoaren posizio zehatz batzuetan trabaturik geratzen da, eta posizio horretatik irtetea asko kostatzen zaio.

Laburbilduz, *besoserb\_infrag\_2* programarekin lortu da besoa aginte infragorriarekin kontrolatzea. Bestalde, lorturiko emaitzak espero zirenak baino txarragoak izan dira, ikusi denez besoa desegonkorra delako. Serbomotorre pare handiagoa dutenez indar handiago dute. Honek, besoa mugiarazterakoan kolpeka dabilela dirudi. Gainera, posizioan mantentzeko agertzen diren oszilazio horien ondorioz besoko doitasuna eta zehaztasuna galtzen da. Hala ere, bere alde onak ere baditu esaterako besoaren pintzaren konexioa zuzenean egiten da ez da kanpoko osagairik behar. Pintzako DC motorra, DC driverrera konektatzen da eta hortik 2. pinera konektatuko da. Hau egitea posible da serbomotoren driverrak bi motore independente kontrola litzakeelako.

## 3.6. OINARRIAREN ETA BESOAREN BATZEA

Dagoeneko robotaren oinarria eta beso biak eraikita daude. Memoria txostenean, alternatibean atalean azaltzen den bezala bi besoak garaturik daudenean beraien ezaugarriak aztertuko dira eta proiektura hobeto moldatzen dena aukeratuko da. Beraz, atal honetan zehar lehenengo beso bien ezaugarri eta funtzionamendua aztertuko da eta beso bietako bat aukeratuko da. Ondoren, aukeraturiko besoa eta oinarriaren arteko batze prozesua bai muntaketa baita programazioa deskribatuko da. Azken atal honen bukaerara heldzean, proiektuan ezarritako helburuak betetzen direla egiaztatuko da.

### 3.6.1. DISEINUA

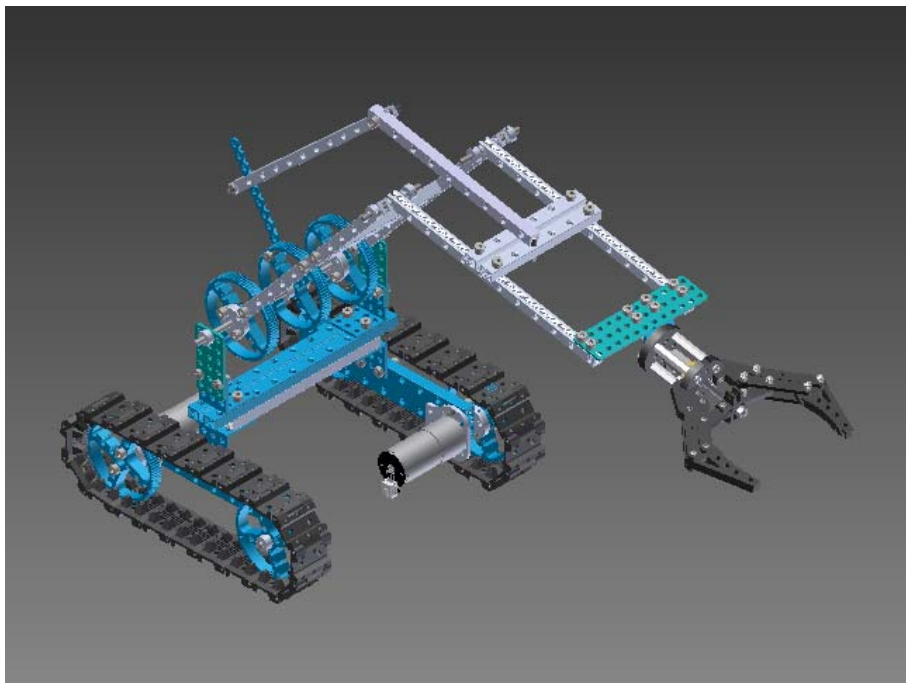
Diseinuarekin jarraitu baino lehen, memoriako alternatibean atalean azaltzen den bezala, besoak garaturik daudenean hauen ezaugarriak aztertu behar dira bietako bat aukeratzeko. Beraz, ondorengo lerroetan behin besoak eraikita eta funtzionatzen daudeneko ezaugarriak aztertuko dira. Izan ere, puntu honetara iristerakoan bietako bat aukeratu beharra dago robota diseinatu eta garatu ahal izateko.

Alde batetik, besoaren egitura hasieratik zutik dago motorrak martxan jarri behar gabe, hau behar beharrezko ezaugarria izango da oinarria higitzen hasten den unerako. Bestetik, besoaren higidura egonkorra da, mugimendua jarraitua kolpe eta oszilaziorik gabekoa. Desabantaila bezala pintzaren funtzionamendu egokia ahalbidetzeko kanpoko driver bat gehitu behar da, honek robotean kokatzeko kable eta osagai gehiago egongo direla esan nahi du. Transmisio zuzeneko besoetan, besoaren egitura eusteko artikulazioetan aurkitzen diren serbomotorrak besterik ez daude, gainera besoa zutik egoteko motorrak martxan egon behar dira. Motorrak, une oro martxan egon behar dira besoaren pisu guztia eusten, motorrek dardarka mugitu egiten dira, ondorioz, beso guztia dar-dar egiten du. Mugimendu hauek, besoaren zehaztasuna eta doitasuna galtzea dakartzate. Horrez gain, serbomotorrek duten indarraren, ondorioz, higitzeko agindua ematean motorrak kolpeka dabilela ikusi da. Beste alde, transmisio zuzeneko besoetan besoko pintza plakara konektatu daiteke kanpoko osagairik erabiliko gabe, nahikoa izango da DC motorren driver bat erabiltzea.

Beraz, bi besoen emaitzak kontuan izanda, beso robotiko mugikorra eraikitzeko robot tradizionala aukeratzea erabaki da. Hainbat aspektutan robot hau egonkorragoa dela ikusi delako, eta hau beharrezko ezaugarria da robot batean.

Beraz, beso tradizionala eta oinarria batu beharko dira. Hau lortzeko besoaren U itxurako euskarriak duen habea eta oinarriaren goi aldeko habea erabiliko dira. Batuketa egin ondoren ziurtatu egin beharko da besoaren mugimenduak ez duela oinarria desorekatzen eta lurrera erorarazten. Horrela bada, besoaren kokalekua mugitu beharko da edota kontrapisua jarri beharko zaio. Bietako edozein aukeratuta ere, kontuan eduki behar da desorekatzen den norabidearen aurkako noranzkoan ipiniko dela beso edo kontrapisua.

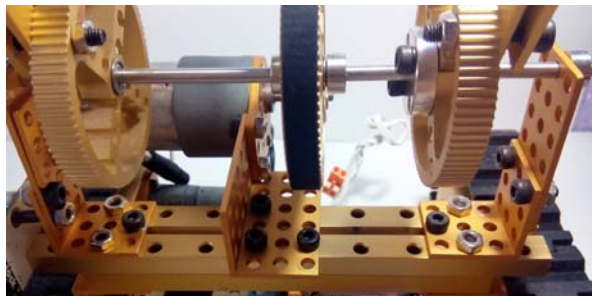
Robotaren zati biak batu ostean, osagai elektronikoak kokatzeko leku bat eraiki beharko da. Toki horretan pilak, plaka eta plakako konexioak egiteko leku nahikoa egotea beharrezkoa izango da. Bestalde, ultrasoinu sentsoreentzat ere leku egokia aurkitu beharko da, kontuan izan behar da aurreko objektu edo oztopoak detektatu behar dituela eta ez robotaren atalen bat (besoa edo pintza) edo lurra.



*Irudia 3.6.1. Beso robotiko mugikorraren 3D-ko diseinua*

### 3.6.2. MUNTAKETA

Muntaketa atalean, robotaren bi zatien batzea deskribatuko da. Besoa, zuzenean oinarriaren gainean ipintzen ahaleginduz gero ezinezkoa izango litzateke, honen oinarrian torloju eta azkoin asko daudelako. Guzti hauek askatuz gero besoa desmuntatuko litzateke, izan ere, torloju eta azkoin hauek motorrak besora finkatzen dituzte eta motorrek kate-mailak. Beraz, aukerarik seguruenak, bien batuketa egin orduko besoa beheko habera eustea da. Habearen behe aldera konektaturiko xafla bakoitza eta habearen errail harilkatuaren artean, M4x8 Allen torloju bat edo pare bat sartuko dira, horrela, beste torloju eta azkoinak askatzen direnean besoa ez da eroriko.

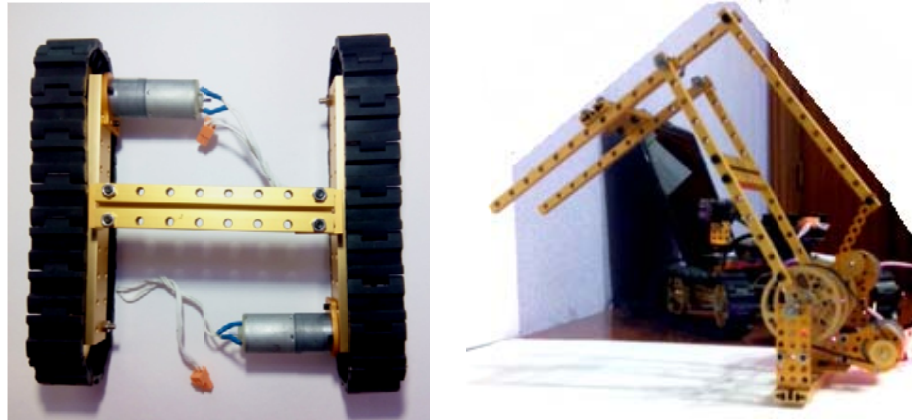


*Irudia 3.6.2. Besoaren euskarria finkatzen*

Nahiz eta besoa finkatuta egon, oinarri eta besoaren egiturak handiak dira pertsona bakar batek maneiatzeko, horregatik, atalka batzen joango dira. Hasteko, robotaren oinarriaren alde biak batzen dituen habea askatuko da. Oinarritik askatutako habea 8 zulokoa da eta besoaren euskarrian dagoenak 10 zulo ditu, baina, habeen tamaina desberdinak izateak ez du inongo arazorik suposatuko, habe biak zentratzearekin nahikoa izango da. Robotaren bi zatien batuketa garatzeko torlojuak bakarrik erabiliko dira. Besoaren U egiturako euskarriko M4x14 allen torlojuak askatu egingo dira eta M4x22 allen torlojuengatik aldatu behar dira, beraz, zazpi M4x22 Allen torloju beharko dira. Kontu handiarekin besoaren euskarriko torloju eta azkoinak askatuko dira. Guztiak askatu direnean, 10 zuloko habearen behean oinarritik askatu den 8 zuloko habea kokatzen da. Batzearekin hasteko, besoaren euskarri erditik kenduriko torlojuak ipiniko dira baina oraingoan M4x22 allen torlojuak azkoinekin jarriko dira, hau egin ostean bi habeak baturik geratuko dira. Beraz, orain oinarriko alde biak batzea besterik ez da geratzen, horretako, 10 zuloko habearen izkinetako bigarren zuloak eta



oinarriaren aldeak bat egiten duten zulotik bi M4x22 allen torloju sartuko dira eta bi azkoinekin batu.

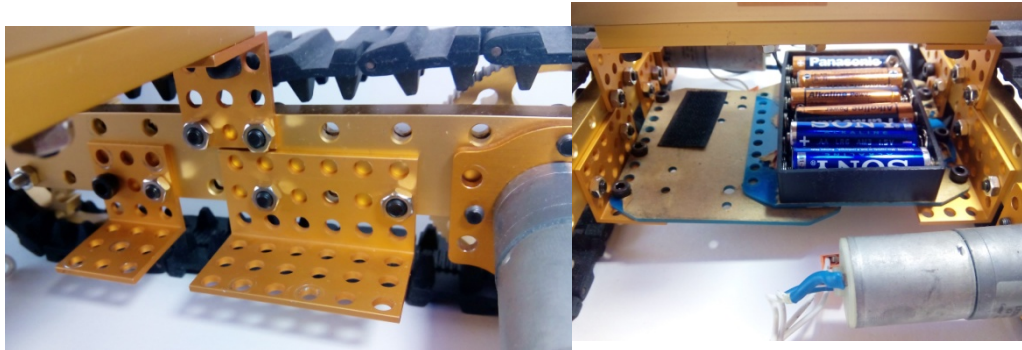


*Irudia 3.6.3. Batuko diren oinarri eta besoa*

Guzti hori egin ostean robota bere osotasunean muntatuta edukiko da. Baina, orain arazoa pila, plakak eta konexioak egiteko lekua behar dela da. Osagai guzti hauek kokatzeko dagoen toki bakarra besoaren azpian eta oinarriaren alde bien artean dagoena da. Bertan, alde batetik bestera doan apal bat eraikiko da, beharrezko materiala hurrengoak dira: 3x3ko L bi xafla, 3x6ko L bi xafla, hamar azkoin eta hamar M4x14 allen torloju. Oinarriaren alde bakoitzean 3x3ko L xafla bi torloju eta azkoinekin eta 3x6ko L xafla hiru torloju eta hiru azkoinekin kokatuko dira. Xafla hauen bigarren ilarako zuloak eta gurpilak batzen dituen habearen beheko zuloak bat egiten duten tokitik sartuko dira torlojuak. Torloju eta azkoin guztiak ondo estutu behar dira, oinarriaren mugimenduari ez daitezela askatu eta osagai guztia bidetik ez galtzeko.

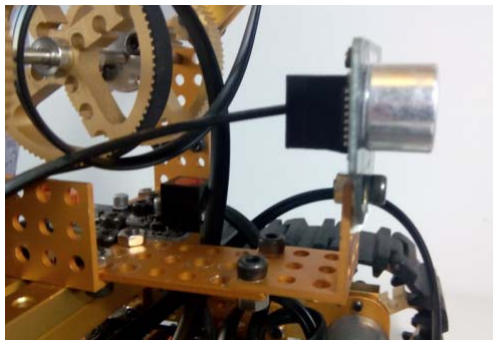
Behin alde biak kokaturik daudenean, apal bat sortzeko bi aldeak batzen dituen zerbait beharko da. Oinarriaren izkina batetik bestera habeak kokatzea ez litzateke ideia ona izango. Izan ere, habeek behar duten leku guztiaz gain hauek lotzeko erabili beharko liratekeen torloju eta azkoin guztiek materiala jartzeko oztopo handia izango lirateke. Hori dela eta, oinarriaren alde batetik bestera heltzen den, leku gutxi hartzen duen eta torloju eta azkoin gutxirekin egonkorra izango den zerbait behar da. Ezaugarri guztiak betetzen dituen material bakarra plaka euskarria da. Hala ere, plaka euskarria ez da alde batetik bestera helduko, horregatik, alde batean plaka euskarria eta bestean pila euskarria bere kaxarekin ipiniko dira. Horrela, lekua izateaz gain, guztia ondo antolatuz garatuko da. Euskarriak, alde bakoitzera ondo

eusteko hiru M4x14 allen torloju eta hiru azkoin erabiliko dira, asko estutu behar dira euskarriak ez direlako alde batetik bestera heltzen eta beraien arteko egonkortasuna lortzeko.



*Irudia 3.6.4. Beheko apalaren eraikuntza*

Robotaren muntaketarekin bukatzeko ultrasoinua kokatzea falta da. Robotaren egitura aztertu ondoren, ultrasoinua kokatzeko bi aukera daudela ikusi da. Lehenengoa, eraiki berri den beheko pisuan kokatzea da. Baina honek izan dezakeen arazoa zera da, pisu hau lurretik oso hurbil dagoenez, ultrasoinuak lurra detektatzea oztopo bat balitz bezala izango litzateke, ondorioz robota gelditu egingo litzateke. Bigarren aukera, batzen diren habearen gainean kokatzea da. Hanean jartzean izan daitekeen arazoa, engranajeekin toki nahiko ez izatea izango litzateke, honen konponbidea habeetatik ultrasoinua kanporantz ateratzea da. Azken aukera hau hautatu da, lehenengo aukera ez delako bideragarria. Hala ere, aukera honek eduki dezakeen arriskua, ultrasoinuak besoa detektatzea eta ondorioz oinarria geldiaraztea izango litzateke. Baina arazo hau edukiko den ala ez, ezin izango da jakin robotaren programazio nagusia garatu arte. Ultrasoinua kanporantz atera ahal izateko egitura eraikitzeko beharrezko materiala 3x6ko xafla bat, 3x3 L xafla bat, sei M4x8 allen torloju eta sei azkoin dira. 3x6ko xafla eta 3x3 L xafla batu egingo dira bakoitzaren lehenengo zuloetatik, zulo horietan bi M4x8 allen torloju sartuko dira eta azkoinekin estutu. Ondoren, ultrasoinua 3x3 L xaflaren goiko aldean ultrasoinua kokatuko da bi M4x8 allen torloju eta bi azkoin erabiliz. Bukatzeko, egitura hau oinarria eta besoa batzen diren habera kokaturiko 3x6 L xaflan lotuko da M4x8 allen torloju eta bi azkoin erabiliz. Egiaztatu behar da ultrasoinua ondo finkaturik dagoela, bibrazioek ez eragiteko.



*Irudia 3.6.5. Ultrasonu sentsorea bere egituran*

Une honetan, robota eraikita egongo da, falta den bakarra osagai elektronikoak ipintzea izango litzateke. Beheko aldean, pila kaxaren ondoan plakak jarriko dira. Plakan beharrezko konexioak egin eta ahalik eta modurik antolatuenan driverrak eta kableak robotean zehar kokatu beharko dira, ez dezaten robotaren mugimendurik eragotzi.



*Irudia 3.6.6. Robot guztia eraikita*

### 3.6.3. PROGRAMAZIOA

Programazioari dagokionez, besoaren eta oinarriaren arteko konbinazio bat eraiki beharko da. Robotaren atal bakoitzean programazio egiterakoan erabilitako agente infragorriko botoi, konexio eta izen desberdinek programa bateratzerakoan sortu daitezkeen nahastean ekidingo dituzte. Ondorengo tauletan plaka konexioak eta agente botoi guztiak bateratu dira programazio nagusia garatzerakoan argi edukitzeko:

<i>Taula 3.6. 1. Roboteko konexioak</i>		
KONEXIOA	IZENA	DESKRIBAPENA
M1	m_esk	Oinarriaren eskumako motorraren konexioa
M2	m_ezk	Oinarriaren eskumako motorraren konexioa
PORT_1	m_goi	Besoko goiko artikulazioko motorraren konexioa. DC motorra eta plakaren artean driverra kokaturik egongo da
PORT_2	m_behe	Besoko beheko artikulazioko motorraren konexioa. DC motorra eta plakaren artean driverra kokaturik egongo da
PORT_3	infragorri_hartz	Infragorri hartzailea konexioa da.
PORT_4	ultras_sents	Ultrasoinu sentsorearen konexioa da.

<i>Taula 3.6.2. Aginte infragorriko kontrola</i>		
AGINTEKO BOTOIA	KONTROLATUKO DUEN ROBOTAREN ATALA	EGINGO DUEN AKZIOA
+	Oinarria	Aurrerantz joan

-	Oinarria	Atzerantz joan
>>	Oinarria	Eskumara biratu
<<	Oinarria	Ezkerrera biratu
1	Besoaren goiko artikulazioa	Gorantz higitu
4	Besoaren goiko artikulazioa	Beherantz higitu
2	Besoaren beheko artikulazioa	Gorantz higitu
5	Besoaren goiko artikulazioa	Beherantz higitu
3	Pintza	Ireki
6	Pintza	Itxi
8	Oinarria	Aldendu

*Prog\_nagusia* programaren helburua, robotaren higikortasuna aginte infragorria eta ultrasoinu sentsorea erabiliz egitea da. Erabiltzaileak, robota higienaraziko du aginte infragorriko goialdean kokaturiko botoiekin, baina, ultrasoinuak 30 cm-tara oztopo edo objektu bat dagoela detektatzen duenean robota geldiaraziko du. Nahiz eta, erabiltzaileak, aurrera jarraitzeko edo mugiarazteko beste edozein agindu eman ~~arren~~, ez dio kasurik egingo eta bere oztopoen aurka ez jotzeko egoera horretan jarraituko du. Ultrasoinuak bere aurrean zerbait dagoela detektatu du, baina, sentsore honek ez daki aurrean dagoen hori oztopo bat edo hartu behar duen objektua den. Beraz, erabiltzaileak zer den ikusiko du eta honen arabera aginte infragorriko botoi desberdinei emango die. Robotaren aurrean dagoena oztopo bat baldin bada, aginteko 8 botoiari ematean robota oztopotik aldentuko da, behin oztopotik aldentu denean eta ultrasoinuak ez duenean beste ezer detektatzen erabiltzaileak oinarria gidatzen jarraitu ahal izango du. Ultrasoinuaren aurrean maneatzeko objektu bat baldin badago, oinarria geldik jarraitzen duen bitartean besoa higienaraziko du hau 1, 2, 3, 4, 5, eta 6 botoiak erabiliz, lehenengo hirurak besoa igotzeko eta pintza irekitzeko eta bestean besoa

jaisteko eta pintza ixteko izango dira. Behin robotak objektua hartu duenean eta ultrasoinutik alden du oinarria gidatu ahalko du objektua lekuz mugitzeko.

```

/*****
Programaren izena: prog_nagusia
Egilea: Itziar Aldekoa
Deskribapena: beso robotiko mugikorra programa nagusia garatuko da
Erabiltzaileak robota mugimendua kontrolatuko du aginteko goi alde
botoiekin. Baina, gg robotak ultrasoinuaren bidez ostoporen bat aurk
ez dio erabiltzaileari kasurik egingo eta geldirik jarraituko du. O
erabiltzaileak bi aukera izango ditu. 1- Robotaren aurrean dagoena
baldin bada 8 botoiari eman eta bertatik aldenduko da. 2-Aurrean d
maniatzeko objetu bat baldin bada beso mugituko du 1,2,3,4,5, eta
botoiekin.
*****/
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeDCMotor m_esk(M1); // M1 sarrera eskumako motorrarena izango da
MeDCMotor m_ezk(M2); // M2 sarrera ezkerreko motorrarena izango da
MeDCMotor m_goi(PORT_1); // 1. pinean goiko kate-maila mugitzeko mot
MeDCMotor m_behe(PORT_2); // 2. pinean beheko kate-maila mugitzeko mot
MeInfraredReceiver infragorri_hartz(PORT_3);
// infragorri hartzailea 3. portuan konektatu
MeUltrasonicSensor ultras_sents(PORT_4); // ultrasoinua 4. portura
MePort pintza(PORT_6); // Pintzake RJ modulura konektaturik

byte motor_abid = 140; // motorren abiadura, PWM-a balio maximoa

void setup()
{
    infragorri_hartz.begin();
}

void loop()
{
    if((infragorri_hartz.buttonState()) && (ultras_sents.distanceCm() >
    {

```

*Irudia 3.6.7. prog\_nagusia programa (1.zatia)*

```

switch(infragorri_hartz.read())
{
    case IR_BUTTON_PLUS:// mandoko + botoia aurrera mugitu
        Aurrera();
        break;
    case IR_BUTTON_MINUS:// mandoko - botoia atzera mugitu
        Atzera();
        break;
    case IR_BUTTON_NEXT:// mandoko --> botoia eskumara bir
        Eskumara();
        break;
    case IR_BUTTON_PREVIOUS:// mandoko <-- botoia ezkerrera
        Ezkerrera();
        break;
}
}
else if((infragorri_hartz.buttonState())&&(ultras_sents.distanceC
{
    switch(infragorri_hartz.read())
    {
        case IR_BUTTON_8:// 8 botoia eman, objektutik aldendu
            Aldendu();
            break;
        case IR_BUTTON_1:// 1 botoia eman, goiko kate-maila go
            Gorantz1();
            break;
        case IR_BUTTON_4:// 4 botoia eman, goiko kate-maila beh
            Beherantz1();
            break;
        case IR_BUTTON_2:// 2 botoia eman, beheko kate-maila go
            Gorantz2();
            break;
        case IR_BUTTON_5:// 5 botoia eman, beheko kate-maila beh
            Beherantz2();
            break;
        case IR_BUTTON_3:// 3 botoia eman, pintza irekiko da
            pintza_irek1();
            break;
        case IR_BUTTON_6:// 6 botoia eman, pintza itxiko da
            pintza_itxi();
            break;
    }
}
else
{
    Stop();
}
}

```

*Irudia 3.6.8. prog\_nagusia programa (2.zatia)*

```

//Goian izendatutako botoiari sakatzean exekutatu diren programa
void Aurrera()// AURRERA MUGITU, motor biak biratzen
{
    m_esk.run(motor_abid);
    m_ezk.run(motor_abid);
}

void Atzera()// ATZERA MUGITU, motor biak beste aldera biratzen
{
    m_esk.run(-motor_abid);
    m_ezk.run(-motor_abid);
}

void Ezkerrera()// EZKERRERA BIRATU, motor bakoitza alde batera bi
// ezkerrekoa kontrako norabidean
{
    m_esk.run(motor_abid);
    m_ezk.run(-motor_abid);
}

void Eskumara()// ESKUMARA BIRATU, motor bakoitza alde batera bira
// eskumakoa kontrako norabidean
{
    m_esk.run(-motor_abid);
    m_ezk.run(motor_abid);
}

void Aldendu()
{
    m_esk.stop();//GEKDIRIK, Motorrak gelditu
    m_ezk.stop();
    delay(200);//0,2 seg
    m_esk.run(-motor_abid); // ATZERA MUGITU, motorrak biraka
    m_ezk.run(-motor_abid); //kontrako norabidean
    delay(1000); // 1 seg
    m_esk.run(-motor_abid);// ESKUMARA BIRATU ardatzaren gainean,
    m_ezk.run(motor_abid); //horretarako motor bakoitza alde batera
    delay(1000); // 1 seg
}

void Gorantz1()// Goiko kate-maila gorantz mugitu horretarako
{
    m_goi.run(motor_abid); // goiko motorra biratu
    m_behe.stop(); // beheko motorra geldik
}

void Beherantz1()// Goiko kate-maila beheantz mugitu horretarako
{
    m_goi.run(-motor_abid); // goiko motorra beste aldera biratu
    m_behe.stop(); // beheko motorra geldik
}

void Gorantz2()// Beheko kate-maila gorantz mugitu horretarako
{
    m_goi.stop(); // goiko motorra geldik
    m_behe.run(-motor_abid); // beheko motorra beste aldera biratu
}

```

*Irudia 3.6.9. prog\_nagusia programa (3.zatia)*



```

void Beherantz2()// Beheko kate-maila beherantz mugitu horretarako
{
    m_goi.stop(); // goiko motorra geldik
    m_behe.run(motor_abid); // beheko motorra biratu
}
void pintza_ireki()// Pintza irekiko da horretarako
{

    pintza.Dwritel(HIGH); // S1 seinalea aktibatuko da
    pintza.Dwrite2(LOW); // S2 seinalea ez da aktibatuko
}

void pintza_itxi()// Pintza itxiko da horretarako
{
    pintza.Dwritel(LOW); //S1 seinalea ez da aktibatuko
    pintza.Dwrite2(HIGH); //S2 seinalea aktibatuko da
}

void Stop()// robota geldirik
{
    m_esk.stop();
    m_ezk.stop();
    m_goi.stop();
    m_behe.stop();
    pintza.Dwritel(LOW);
    pintza.Dwrite2(LOW);
}

```

*Irudia 3.6.10. prog\_nagusia programa (4.zatia)*

Robot osoaren programaren funtzionamendua egiaztatzerakoan besoa posizio egoki batean egonez gero ez diola ultrasoinuari eragozten egiaztatu da. Bestalde, agian programako datu batzuk doitu egin beharko direla erabaki da, esaterako, motorren abiadura eta ultrasoinuaren distantzia. Motorren abiadura handia bada, hau da, robota oso arin higitzen bada, ultrasoinuari ez dio denborarik emango irakurketa egiteko eta aurreko oztopo edo objektuaren kontra joko luke. Ultrasoinuaren distantziari dagokionez, sentsoreak objektu bat detektatzen duenean, objektua robotetik hurbil gelditzen da, beraz, besoaren luzeraz baliatuz distantzia pixka bat handitu daiteke.

Laburbilduz, *prog\_nagusia* programarekin ez da bakarrik atal honetan ezarritako helburua lortu, baizik eta programa honen inplementazioari esker proiektuan hasiera batean ezarritako helburuak lortu direla egiaztatu da. Beso robotiko mugikor baten diseinu, programazio eta muntaketa garatzea lortu da.