

GRADUA: INDUSTRIA ELEKTRONIKAREN ETA  
AUTOMATIKAREN INGENIARITZA

## GRADU AMAIERAKO LANA

***PILOTA BATEN ALTUERAREN KONTROLA  
ARDUINO ETA MATLAB ERABILIZ***

***1. DOKUMENTUA - MEMORIA***

**Ikaslea:** Uriguen Fernandez, Ekain

**Zuzendaria (1):** Casquero Oyarzabal, Oskar

**Zuzendaria (2):** Armentia Diaz de Tuesta, Aintzane

**Ikasturtea:** 2018-2019

**Data:** Bilbon, 2019-07-17

## LABURPENA

Gaur egun, mikrokontrolagailuek hartzen ari diren garrantziarekin, hainbat era garatu egin dira programak sortzeko, maila baxuko lengoaietatik, maila altukoetara. Gero eta konplexuagoak izaten diren heinean, gero eta maila altuago bat edukiko dute sailkapen simple bat eginez, adibidez hurrengoak daude: makina lengoia, mihizadura lengoia, c lengoia eta ingurune bisuala. Eman den mikrokontrolagailuen zabalkuntza programazio ingurune bisualak garatzea erakarri du, hainbat programazio ingurune bisual berri garatu egin dira, adibidez, Matlab eta Labview. Programa horiekin modelaketa egiten da, hain zuzen kontrol automatikoaren modelaketa. Programazio ingurune horiek, programazioa errazten dute. Proiektu honetan, PID kontrol bat garatuko da, pilota baten altuera haizagailu baten bitartez kontrolatzeko, horretarako bi gauza aztertuko dira, Matlab eta Arduino inguruneak ezberdintasunak ikusteko alde batetik, eta beste alde batetik, Matlab –en sensorearen seinalea eskuratzeko aurretik zehaztutako blokeak erabiltzearen eta s-function blokea erabiltzearen ezberdintasunak ulertzeko.

Hitz-gakoak:

Arduino, Matlab, Simulink, PID, s-function.

## RESUMEN

Con la importancia que están obteniendo los microcontroladores hoy en día, se han desarrollado numerosas formas para hacer programas, desde lenguajes de bajo nivel a lenguajes de alto nivel. Cuanto mayor nivel tengan tendrán mayor complejidad, haciendo una clasificación simple, por ejemplo tenemos los siguientes: Lenguaje máquina, ensamblador, lenguaje en c y entorno visual. Con la difusión que se ha dado de los microcontroladores, se ha logrado desarrollar nuevos entornos visuales de programación, como por ejemplo, Matlab y Labview. Con esos programas se hace modelado, concretamente modelado de control automático. Esos entornos facilitan la programación. En este proyecto se va a desarrollar un control PID para controlar la altura de una bola con un ventilador, para ello se van a comprobar dos cosas, por una parte los entornos Matlab y Arduino, para ver las diferencias entre uno y otro, por otra parte se van a comprobar los modos de captura de señal en Matlab para entender las diferencias de utilizar los bloques predefinidos en Matlab o utilizar el bloque s-function.

Palabras clave:

Arduino, Matlab, Simulink, PID, s-function.

## ABSTRACT

The microcontrollers are getting a lot of importance today; there are numerous ways to make programs that have been developed, from low-level languages to high-level languages. The higher the level, the more complex they will be. A simple classification, for example would be the following: Machine language, assembler, c language and graphic environment. With the diffusion of microcontrollers, it has been possible to develop new graphic programming environments, such as Matlab or Labview. Modeling can be made with these programs, specifically control modeling. These environments facilitate programming. In this project we are going to develop a PID control to control the height of a ball with a fan. In order to make this happen two things are going to be checked, on the one hand the Matlab and Arduino environments, to see the differences between one and the other, on the other hand the signal capture modes will be checked in Matlab to understand the differences of using the predefined blocks in Matlab or using the s-function block.

### Keywords:

Arduino, Matlab, Simulink, PID, s-function.

## AURKIBIDEA

1.	SARRERA.....	1
2.	TESTUINGURUA.....	2
3.	HELBURUAK.....	3
4.	ONURAK .....	5
5.	BALDINTZEN DESKRIBAPENA.....	6
6.	ALTERNATIBEN ANALISIA .....	7
6.1.	Mikrokontroladorea.....	7
6.2.	Distantzia sentsorea.....	8
6.3.	Eragingailua.....	9
7.	PROPOSATUTAKO DISEINUA .....	10
8.	BEHE MAILAKO DISEINUA .....	14
8.1.	Hardware Atala .....	14
8.1.1.	Datuaren eskuraketa.....	14
8.1.2.	Prozesaketa .....	16
8.1.3.	Irteerako seinalearen sorketa .....	21
8.1.4.	3D Inpresioa .....	22
8.1.5.	Konexioen Eskema .....	24
8.2.	Software Atala.....	25
8.2.1.	Arduino.....	25
8.2.2.	Matlab – Simulink.....	31
9.	PLANGINTZA.....	43
9.1.	Proiektuaren partaideak .....	43
9.2.	Proiektuaren paketeak.....	43
9.2.1.	Gestioa .....	43
9.2.2.	Prestakuntza.....	44
9.2.3.	Formakuntza.....	45
9.2.4.	Frogapena.....	46
9.2.5.	Dokumentazioa .....	48
9.3.	Gantt Diagrama .....	49
10.	AURREKONTUA.....	51
11.	ONDORIOAK .....	53
12.	BIBLIOGRAFIA.....	54
	ERANSKINAK 1: Arduino -aren barne eskema.....	56

ERANSKINAK 2: Kodea eta Parametroak.....	57
ERANSKINAK 3: Planoak .....	58

## Irudien Zerrenda

Irudia 7-1 Plantaren Irudia .....	10
Irudia 7-2 Arduino Mega 2560 .....	11
Irudia 7-3 Begizta itxiko sistema .....	11
Irudia 7-4 Sistemaren eskema elektrikoa.....	12
Irudia 8-1 Sharp GP2Y0A21YK sentsorea .....	14
Irudia 8-2 Irteera boltaia vs distantzia kurba .....	15
Irudia 8-3 Doiketa kurba.....	16
Irudia 8-4 Arduino Mega 2560 .....	17
Irudia 8-5 PID kontrolagailuaren eskema eta ekuazioak.....	20
Irudia 8-6 PWM seinalearen adibideak .....	20
Irudia 8-7 COM-11270 eragingailua .....	21
Irudia 8-8 Prusa i3 inprimagailua .....	22
Irudia 8-9 Airea eskuratzen duen oinarriaren 3d ereduaren atala .....	23
Irudia 8-10 Oinarriaren 3d ereduaren atal trinkoa .....	23
Irudia 8-11 Eragingailuaren eta pilotaren tutuaren konexioa.....	24
Irudia 8-12 Konexioen eskema.....	24
Irudia 8-13 Mikrokontrolagailua eta Arduino IDE -aren arteko interakzioa .....	25
Irudia 8-14 Arduino programa baten atal nagusiak .....	26
Irudia 8-15 Arduino programaren egitura.....	27
Irudia 8-16 Buffer zirkularraren azalpena .....	28
Irudia 8-17 Eulerren lehenengo hurbilketa .....	29
Irudia 8-18 Eulerren bigarren hurbilketa .....	29
Irudia 8-19 Hurbilketa Trapezoidala.....	30
Irudia 8-20 External mode.....	31
Irudia 8-21 Matlab eta Arduinoaren arteko elkarrekintza .....	32
Irudia 8-22 Bloke sistemaren programa.....	33
Irudia 8-23 S-function bloke sistema .....	33
Irudia 8-24 Sentsorearen seinalearen bihurtzea .....	34
Irudia 8-25 s-function exekuzio modu orokorra .....	35
Irudia 8-26 s-function builder .....	36
Irudia 8-27 Step funtzioaren sorketa .....	36
Irudia 8-28 Kode sorketaren output funtzioa .....	37
Irudia 8-29 s-function -ren funtzionamendu diagrama.....	38
Irudia 8-30 S-function hasieraketa atala .....	38
Irudia 8-31 S-function irteeren atala.....	39
Irudia 8-32 S-function liburutegiak atala.....	39
Irudia 8-33 S-function eguneraketaren atala .....	40
Irudia 8-34 S-function main atala.....	41
Irudia 8-35 PID blokearen xehetasunak .....	42
Irudia 8-36 PWM irteera .....	42

## Taulen Zerrenda

Taula 6-1 Mikrokontrolagailuaren aukeraketaren balorazioa .....	8
Taula 6-2 Distantzia sentsorearen aukeraketa.....	8
Taula 6-3 Eragingailuaren aukeraketa.....	9
Taula 8-1 Doiketa kurba egiteko datuak .....	15
Taula 8-2 Arduino Megaren ezaugarriak.....	17
Taula 8-3 Eragingailuaren ezaugarriak .....	21
Taula 9-1 Lan paketeen laburpena.....	49
Taula 9-2 Lan eskakizunen garrantzia .....	49
Taula 9-3 Gantt Diagrama .....	50
Taula 10-1 Garapen Kontzeptualaren Aurrekontua.....	51
Taula 10-2 Proiektuaren Ezarpen Aurrekontua .....	52
Taula 10-3 Kostu Totala.....	52
Taula 0-1 Inprimaturiko piezen planoen sailkapena .....	58



## Akronimoen Zerrenda

PID: Proportional-Integral-Derivative controller.

ADC: Analog Digital Converter.

PWM: Pulse Width Modulation.

SRAM: Static Random Access Memory.

EEPROM: Electrically Erasable Programmable Read-Only Memory.

RPM: Revolutions Per Minute.

CFM: Cubic Feet per Minute.

CAD: Computer Aided Design.

STL: Standard Triangle Language.

TTL: Transistor - Transistor Logic.

ms: milliseconds.

IDE: Integrated Development Environment.

FDM/FFF: Fused Deposition Modeling.

## 1. SARRERA

XXI. mendean informatikaren implementazioa ia totala da, bizitzako hainbat alderdietan implementaturik dago, hala nola, etxeetan, industrian, eskoletan, automobilgintzan.... Nahiz eta implementazioa oso handia izan, erabiltzaileen gaitasunak ez dira aurreratu hain abiadura handian.

Informatikaren eta elektronikaren garapenak azken urteotan aurrerapen asko ekarri ditu, oso txikiak diren eta gaitasun handiko mikrokontrolagailuen sorketa ahalbidetuz, baita ere, gero eta merkeagoak dira.

Mikrokontrolagailuek gero eta gaitasun handiagoak dauzkate eta era errazago batean programatzen dira, horrek mundu hau jendarte anitzera hurbiltzea dakar, era horretan, hainbat aplikazio edonork egin ditzake. Horretarako hainbat baliabide garatu dira azken urteotan, adibidez; Matlab suitea non bloke sinpleen bitartez programazio ingurune bisual bat sortu daiteke era erraz batean, modu horretan aplikazio asko garatu daitezke c programazioa menperatu barik.

Azken urteotako aurrerapenei esker, Matlab –ek zuzenean elkarri eragiten du Arduino -arekin, horrela Matlab –ek, programazioa burutzen du Arduino -aren barnean balioak zuzenean aldatuz, aldi berean denbora errealean balioen aldaketa ahalbidetzen du. Horregatik denbora laburrean frogak egin daitezke, balioen eskurapena eta aldaketa momentuan eginez, doiketa denborak asko murriztuz.

Aurrerapen honek kontrol sistemetan pausu handi bat izan da, alde batetik, mikroprozesagailuen aurrerapenengatik eta bestetik programazioaren baliabide berriengatik. Gaur egungo kontrol sistemak gero eta gaitasun handiagokoak dira eta sinpleagoak, hori dela eta proiektu honetan desberdintasun horiek aztertuko dira.

## 2. TESTUINGURUA

Garai honetan, mikrokontrolagailuek duten garrantzia gero eta handiagoa da, gero eta eragin handiagoa dauzkate eguneroko bizitzaren edozein arlotan, baita ere, kontrol automatikoaren garapena ahalbidetzen dute hainbat aplikazioetarako programaturiko edozein kodearekin.

Gaur egun, mikrokontrolagailuen erabilera dela eta, hardware eta software librean oinarritzen direnak, interakzioa egiteko modua gero eta arlo gehiagoetara hedatu da askoz errazagoa delako, alde batetik, denbora errealean aldagaien balioen kudeaketa zuzena egin daiteke, frogak egiteko denbora asko aurreztuz, era horretan errendimendua askoz handitzen da, beste alde batetik, eragingailuaren seinale eskuraketa eta prozesaketa ahalmena eraginkorragoa Matlab -en, beraz, prozesuaren abiadura eta zehaztasuna askoz handiagoa izango da.

Helburua proiektu honetan, Matlab -eko blokeak eta Matlab -en c kodea txertatzearen arteko eraginkortasuna zenbat aldatzen den aztertzea eta baita ere, programazio ingurune bisuala eta kode bidezko programazioaren desberdintasunak eta onurak aztertzea, PID kontrol baten doiketa eginez.

Prozedura hauek azaltzeko, kasu praktiko bat garatzea erabaki da. Kasurako pilota baten altueraren kontrola PID kontrolagailu baten bitartez egingo da, horretarako erabiliko den eragingailua haizagailu bat izango da. Hori Matlab eta Arduino -aren bitartez garatuko da.

### 3. HELBURUAK

Helburua Gradu Amaierako lan honetan, haizagailu batek eragindako pilota baten altuera kontrolatzea da, PID baten bitartez. Lan honetan, pisu txikiko pilota bat mugituko da, dena den, pisu handiagoko pilota bat erabili ahal da, ekipo egokia aukeratuz.

Horretarako, sentsore baten bitartez, pilotaren altuera neurtuko da, PID -ari informazioa bidaliko zaio, berak distantzia hartuko du eta boltaia bihurtuko du. Tentsio hori PID -ak aldatzen du, desiratutako posizioa lortu arte, eragingailuari tentsio berria emanez, pilota orekatu arte. Horrela pilota orekaturik geratuko da. Azaldutako funtzionamendua, Erregulazio Automatikoaren eta Kontrolaren arloari dagokio, hori dela eta, graduko (Ingeniaritza Elektronikoa Eta Automatikoa) konpetentziak garatuko dira lan honetan.

Inplementatzeko, Arduino Mega bat erabiliko da irteerak/sarrerak kontrolatzeko, baita ere programazioa burutzeko, kasu batean, beste batean, ordenagailua erabiliko da, eragingailu bat pilota igotzeko eta sentsore bat posizioa neurtzeko.

Lan hau, bi modu ezberdinetan egingo da, alde batetik, Arduino batean programa sartuz, eta beste aldetik, Matlab erabiliz.

Arduino soilik:

Kasu honetan, seinale analogikoa eskuratuko da, programa baten bitartez prozesatuko da (Arduino -aren barnean) eta eragingailuari PWM seinale bat emango zaio.

Matlab erabiliz:

Hemen, Arduino -aren irteerak eta sarrerak erabiltzen dira, baina programatzeko Matlab erabiltzen da ordenagailutik zuzenean. Horretarako, Matlab garatu du programa bat Arduinoan sartzeko eta programa horrekin zuzenean Matlab erabiltzea ahalbidetzen da.

Bi aukera bereizten dira:

- Simulink –en aurretik zehaztutako blokeen bitartez egindako kontrola, eskuzko funtzionamendua eta funtzionamendu automatikoa ahalbidetzen duena.

- Simulink –en s-function funtzioa erabiliz, sentsorearen atalaren funtzionamendua, hau da, seinalearen eskuraketa eta prozesaketa c kode bitartez exekutatu.

Lan honek izango duen eragina, graduan garatutako gaitasunak aplikatzea eta bermatzea da, urte hauetan ikasitako guztia aplikatuz.

## 4. ONURAK

Proiektu honetan bi gauza aztertuko dira, alde batetik, programazioaren ingurunea eta bestetik, seinale eskuraketa eta prozesaketa. Bi alderdi hauek batuz, denbora errealean konfiguratu daitekeen kontrol sistema bat aztertuko da, ingurune bisualeko programazioa garatuz.

Prestakuntzari dagokionez, denbora luzean landu den gaiaren ezagupen sakona bereganatzea dakar. Graduan landutako ikasgaiekin zerikusia duen proiektu batek, ikasitakoa finkatzea errazten du. Aldi berean, teknika eta atal berriak ikastea laguntzen du, beraz, etorkizunari begira formakuntza osatzeko balio du.

Lan hau aurrera eramatean, seinale analogiko baten eskurapen eta prozesamenduari buruzko ezagupen berriak garatuko dira eta horrek kontrol automatikoan bide berri bat ezagutzea baimenduko du, Matlab –en c kodeak dituen errekurtsoak erabiltzean errendimendua askoz handiagoa izango da Matlab bidez garaturiko blokeen bitartez baino.

Proiektu honetan agertzen diren teknikak hainbat onura dakartzate. Alde batetik lana errazago egitea lortuko da, langileen prestakuntza programazio mailari dagokionez ez da oso handia izan behar eta beste aldetik, froga gehiago egin ahal dira denbora tarte txikiagoan, PID –aren doiketa egiteko, balioak aldatuz eta zuzenean ikusiz sistemaren erreakzioak.

Ekonomiaren aldetik, onura nabarmenak egongo dira. Doiketa denboraren murrizketa dela eta, lan ordu asko aurreztuko dira, aldi berean, langileen prestakuntza ez da hain handia izan behar, beraz, langileen soldatak ez dira hain handiak izango. Eragin hauekin, ikuspegi interesgarria duen proiektua da.

## 5. BALDINTZEN DESKRIBAPENA

Egin behar deneko proiektuaren baldintzen deskribapena atal honetan egingo da, proiektuaren eginkizunak burutzeko asmoarekin. Batzuetan, proiektua garatzean ezinezkoa da helburuak betetzea, garapenean sortu daitezkeen arazoengatik edota, aurreko planteamenduak onak ez izateagatik. Egoera horietan kasua aztertu beharra dago eta irtenbide bat ala beste bat bilatu behar da, denbora eta kostuaren arabera batez ere, baina beti helburua kontuan edukiz.

Bete behar diren baldintzak teknikoak, hauexek izango dira:

- Alde batetik programazio osoa Arduino bitartez egin behar da eta bestetik Matlab erabiliz.
- Distantzia sentsorearen (sentsore analogikoa) neurketaren eskuraketa.
  - Arduino baten programazioaren bitartez.
  - Matlab bloke baten bitartez.
  - Matlab bitartez c kodigoa erabiliz, s-function bitartez.
- Señalea kudeatzeko mikrokontroladore bat erabili, gure kasuan Arduino Mega.
- PID baten inplementazioa, denbora errealean aldaketak jasotzen dituen.
- Eragingailuari ailegatu behar zaion PWM señalea doitu.

Bete behar diren funtzionamendu baldintzak, hauexek izango dira:

- Pilota altuera konkretu batetik beste batera mugitzea.
- Altuera aldaketa bortitzak egitea eta pilota ondo erregulatzea altuera zehatz batean.
- Pilotaren ibilbidea 60 zentimetro ingurukoa izango da.

## 6. ALTERNATIBEN ANALISIA

Proiektua garatzeko hainbat gailu eta osagai ezberdin erabil daitezke, beraz, lana garatzeko hainbat irizpide kontuan hartu behar dira eta erabilpen zehatzaren arabera baldintzen aukeraketa egokia egin behar da. Merkatuan dauden aukera anitzak aztertuz, aplikazio honetarako garrantzitsuak diren alderdiak aztertu behar dira, beraz, helburu berdina duten gailuen artean, prezioa, kontsumoa, memoria, atakak... aztertuko dira eta aukera egokiena, normalean eraginkorrena hautatuko da. Horretarako, dagokion gailuaren atal bakoitzak konparatuko dira garrantzitsuenak diren atalak aztertuz, erabaki egokiena hartzeko asmoarekin. Kasu honetan, Mikrokontroladorea, distantzia sentsorea eta eragingailua aztertuko dira.

### 6.1. Mikrokontroladorea

Aukeraketa egiteko, Arduino Mega eta Arduino Uno –ren ezaugarrien arteko konparaketa egingo da eta beharren arabera bata ala bestea aukeratuko da.

Beharrak:

- **Pin kopurua:** zenbat pin beharko dira?
  - 2 blow fan-arentzeko (PWM eta GND)
  - 3 distantzia sentsorearentzat (Vcc, GND eta Vout)
  - Guztira: **2 pin GPIO** + 2 pin elikadura (Vcc eta GND). Honetarako, Arduino Uno-rekin nahikoa.

Ezaugarriak:

- **Timer kopurua:** mega -k 6 ditu (horietako hiru 16 bit -ekoak direlarik), uno -k aldiz 3 (horietatik 16 bit -ekoa bakarria delarik). Haizagailuaren abiadura kontrolatzeko 16 biteko timer batekin eta 8 biteko beste timer batekin nahikoa da; beraz, Arduino Uno -rekin nahikoa.
- **Kanpo etendura pin kopurua:** mega -k 6 ditu, uno -k aldiz 2. Birekin nahikoa da enkoder birakaria erabiltzeko; beraz, Arduino Uno -rekin nahikoa.



- **Portu serie kopurua:** mega-k 4 portu serie ditu, uno -k aldiz bakarra. Baina ez dira serie porturik behar.
- **Ram memoria:** Mega 8k ditu, aldiz, uno 2k, gero eta memoria gehiago hobeto Matlabekin interakzioa egiteko.

Analisi hau egin eta gero, Arduino Mega erabiltzea komenigarriena dela ondorioztatu da, batez ere, memoria kopuru extra hori duelako eta ordenagailu eta Arduino -aren arteko lan karga hobeto kudeatuko delako.

Taula 6-1 Mikrokontrolagailuaren aukeraketaren balorazioa

MOTA	PIN KOPURUA	PREZIO A	MEMORI A	SERIE PORTU KOP	TIMER KOP	BALORAZIO A
Ponderazioak	15%	10%	40%	15%	20%	
Arduino Mega	10	6	10	8	8	8,9
Arduino Uno	7	8	5	2	6	5,35

## 6.2. Distantzia sentsorea

Lan esparrua metro baten inguruan egongo da, hori dela eta Sharp GP2Y0A21YK sentsorea erabiliko da. Gainerako sentsoreek distantzia esparru ezberdinetan lan egiten dute, VI6180x sentsorea 0,05 – 10 cm tartean lan egiten du, beraz aplikazio honetarako ez du balio eta LV-EZ3 sentsorea 6 metro baino gehiagoko distantziak kudeatu ditzake, beraz, hoberena Sharp infragorri sentsorea erabiltzea da, distantzia esparru horretan lan egin dezake 10-80 cm –ko tartean, bereizmena askoz handiagoa izango da, VI6180x sentsorearena baino.

Sentsore infragorria erabiltzeak, zehaztasun handiagoarekin lan egitera ahalbidetzen du eta baita ere erantzun azkarragoarekin. Aplikazio honetan oso garrantzitsua da, erantzun azkarra edukitzea pilotaren kontrola hobeto egin ahal izateko.

Taula 6-2 Distantzia sentsorearen aukeraketa

MOTA	TAMAINA	PREZIOA	RANGOA	KONEXIO MODUAK	BALORAZIOA
Ponderazioak	15%	35%	35%	15%	
VI6180x	9	6	1	4	4,4
LV-EZ3	5	3	10	9	6,65
GP2Y0A21YK	5	8	9	4	7,3

### 6.3. Eragingailua

Eragingailua aukeratzeko, hainbat ezaugarri aztertu daitezke, adibidez; prezioa, dimentsioak, aire fluxuaren irteeraren diseinua, biraketa abiadura.... Kasu honetan atalik garrantzitsuen irteeraren diseinua da, COM-11270 izeneko eragingailuak irteera konikoa duenez hoberena da, 3D inprimagailuaren zehaztasuna ez denez hain handia, pieza koniko bat ahokatzea errazagoa da.

Prezioaren aldetik merkeena da eta abiadura aldetik ez da behar hainbesteko biraketa abiadura, 3000-3500 rpm baino gehiago ez dira behar pilotaren kontrol egokia gauzatzeko.

Taula 6-3 Eragingailuaren aukeraketa

MOTA	IRTEERAREN				BALORAZIOA
	DISEINUA	PREZIOA	RPM	DIMENTSIOAK	
Ponderazioak	50%	20%	15%	15%	
COM-11270	10	7	6	8	7,56
BFB1012VH	5	4	8	7	5,55
HT-07530D12	5	4	6	7	5,25

## 7. PROPOSATUTAKO DISEINUA

Lehenik eta behin sistema osoaren deskribapena egingo da, proiektuaren irtenbide posibleak aztertuko dira gero ebazpen orokorra azaldu ahal izateko. Hurrengo atalean azalpen sakonago bat emango da proiektua osoa atalez atal deskribatuz.

Proiektua, bi ataletan banatzen da: Hardware eta Software atalak.

Hardwarea, honako elementu hauek gauzatuko dute: Estructura eta oinarria (3D bidezko inpresioa erabiliz), eragingailu bat, hurbiltasun sentso bat eta erabiliko den mikrokontroladorea, Arduino bat kasu honetan.



**Irudia 7-1 Plantaren Irudia**

*Iturria. Berezko Elaborazioa.*

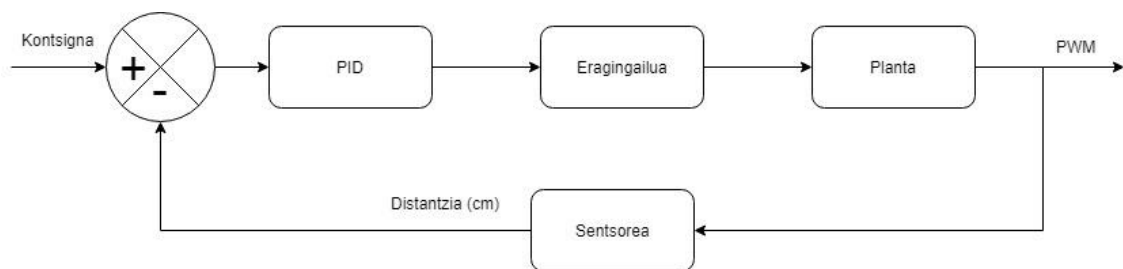
Orain software atala aztertuko da, hemen bi atal bereizten dira: Arduino soilik edo Matlab –en bidezko funtzionamendua.



**Irudia 7-2** Arduino Mega 2560

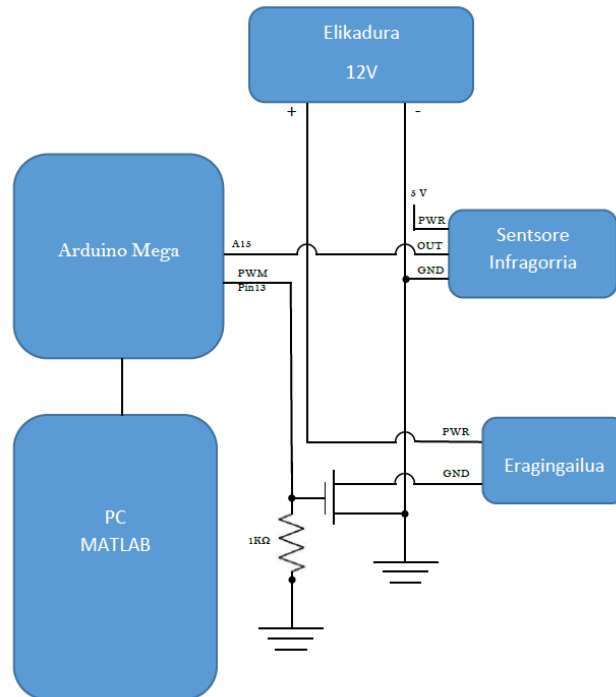
*Iturria.* <https://cutt.ly/ULvuCO>

Bi kasuetan, erregulazio automatikoan ikasitako begizta itxi batean lan egingo da. Sentsoreak lagina hartzen du, Arduino –ari seinalea bidaliko dio eta berak aplikatuko dio kontrola PID –a erabiliz, gero eragingailuari seinalea emango dio prozesu hori behin eta berriro eginez.



**Irudia 7-3** Begizta itxiko sistema

*Iturria.* Berezko Elaborazioa.



Irudia 7-4 Sistemaren eskema elektrikoa

*Iturria. Berezko Elaborazioa.*

Arduino soilik: Hemen Arduino -arekin soilik lan egingo da, programa kargatuko da eta berak egingo du PID –aren funtzionamendua, aldez aurretik sentsoretik jasandako eta eraldatutako sarrera erabiliz, gero PWM seinale bat eragingailuari emanaz.

Matlab bidez: Kasu honetan, Arduino –a baita ere erabiliko da, baina irteerak eta sarrerak kudeatzeko eta programa exekutatzeko bakarrik, programazioa ingurune bisual batean ordenagailuaren bitartez egingo da. Matlab –ek Arduino -arentzat diseinaturiko programa baten bidez, zuzenean programatzea ahalbidetzen duena, modu horretan zuzenean erabili ahal da ordenagailua denbora errealean sistema doitzeko. Matlab –en bidez, PID kontrolagailua diseinatuko da, Kp, Kd eta Ki konstanteak kontrolatuz, horrela pilotaren oreka lortuko da.

Matlab –en barnean, bi froga ezberdin egingo dira, zuzenean sentsoretik lortutako balioa erabiliz, bloke baten bitartez eta S-function bloke baten bitartez, bloke hau c lengoia erabiliz programatuko da.

Bi atal hauek landuko dira eta lortutako emaitzak beraien artean konparatuko dira. Behin lan hori eginda, ondorioak aterako dira eta lanean adieraziko dira.

Arduino -a elikatzeko, ordenagailua erabiliko da edo zuzenean sarera konektatuko da, transformadore baten bitartez.

Sentsorea eta eragingailua, zuzenean Arduino -aren bitartez elikatuko dira.

## 8. BEHE MAILAKO DISEINUA

Aurreko atalean proiektua gainetik deskribatu egin da ikuspegi orokor batetik, atal honetan deskribapena askoz zehatzagoa egingo da atal guztiak sakonki azalduz, hauxe izango da proiektuaren mamia. Proiektu hau bi atal nagusietan banatzen da, Hardware eta Software atalak.

### 8.1. Hardware Atala

Hardware –ari dagokionez, proiektu hau hiru atal nagusietan banatzen da, datuaren eskuraketa, datuaren prozesaketa eta hortik aurrera irteera seinalearen sorketa.

#### 8.1.1. Datuaren eskuraketa

Pilotaren altuera neurtzeko, Infragorritzko Sharp GP2Y0A21YK hurbiltasun sentsorea erabiliko da. Sentsore honek hainbat ezaugarri ditu:

- 10-80 cm tartean lan egiten du.
- Lan tartea 0-3,3 V –koa da.
- Detektatu beharrezko objektuak opakoak eta mateak izan behar dira.
- Ez da behar kanpoko kontrol zirkuiturik.
- Lagin bakoitza hartzeko, 52,9 ms behar dira.
- Merkea.

Konexio pinak, hauek dira:

- V0: Lan tentsioa (5V)
- GND: Lurra
- Vcc: Irteera tentsioa, 0 eta 3.3V artean.

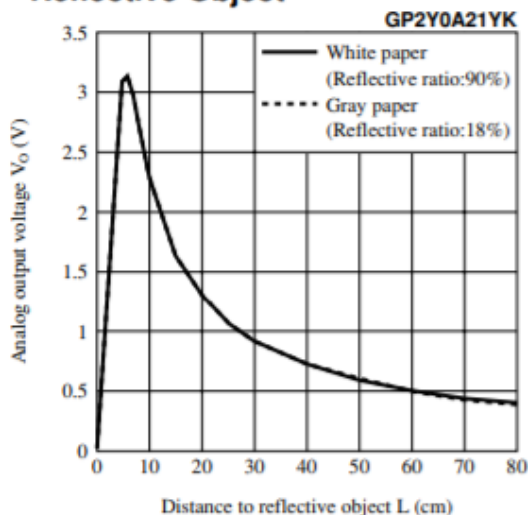


Irudia 8-1 Sharp GP2Y0A21YK sentsorea

Iturria. <https://cutt.ly/dLvdHJ>

Sentsore analogikoa denez, distantziaren eta sentsorearen irteera boltaiaren arteko proportzioa ez da lineala, hori dela eta, datasheet –ean ageri da funtzionamendu kurba, sentsore honen kasuan, 0-10 cm –ko tartean ez da erabilgarria, gero kurba gero eta linealagoa da 10-80 cm –ko tartean, hortik aurrera funtzionamendu egokia ez da bermatzen. Gero eta distantzia luzeagoa neurtzean, irteera tentsioa txikiagoa da.

**Fig.5 Analog Output Voltage vs. Distance to Reflective Object**



Irudia 8-2 Irteera boltaia vs distantzia kurba

Iturria. <https://cutt.ly/yLbcy1>

Datasheet-ean dagoen informaziotik sentsorearen ezaugarri-kurbaren ekuazioa lortuko da, sentsorearekin frogatuz tentsioaren eta distantziaren arteko erlazioa.

Joera zuzena egiteko, hainbat era ezberdin daude, lineala, logaritmikoa, polinomikoa, potentziala eta bataz besteko mugikorra, haztamuka frogatuko da eta errore txikiena ematen duen era hartuko da. 6. Ordenako doiketa polinomikoa egingo da, ia errorerik ez dagoela bermatzen duena.

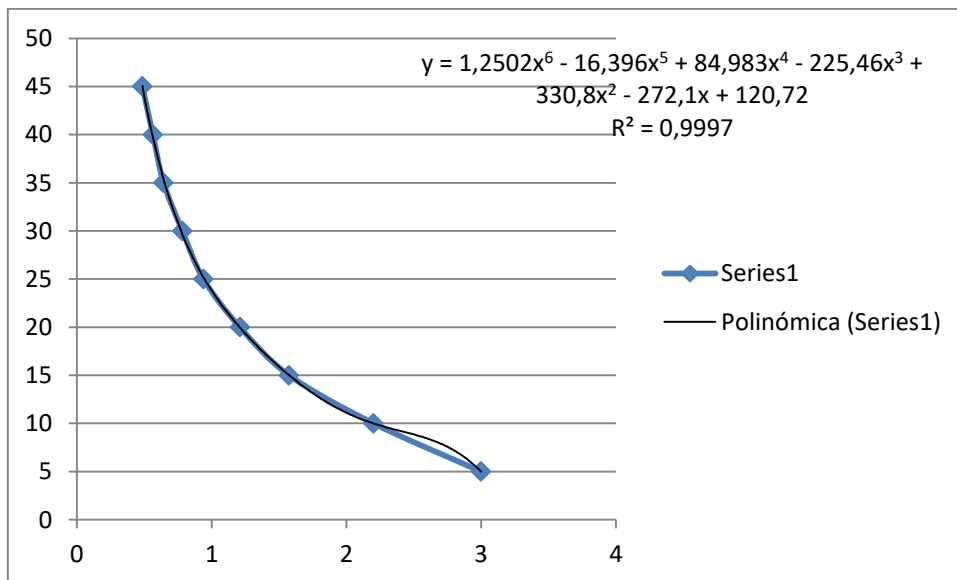
Taula 8-1 Doiketa kurba egiteko datuak

Voltaia [V]	Distantzia [cm]
0,486	45
0,564	40



0,643	35
0,783	30
0,939	25
1,21	20
1,575	15
2,202	10
3	5

Sentsore honek ez ditu emaitza oso zehatzak ematen eta arazo horri aurre egiteko errorea murrizteko ahalik eta gehien doitu da. Gainera, sentsore honek irteera analogikoa ematen du eta Arduino Mega -rekin ezin daiteke zuzenean irakurri. Beraz, sentsorearen irteera Arduino -aren sarrera bezala definitutako ADC pin batera konektatzen da. ADC -a erabili da irteera analogikoa, digitalean bihurtzeko.



Irudia 8-3 Doiketa kurba

*Iturria. Berezko Elaborazioa.*

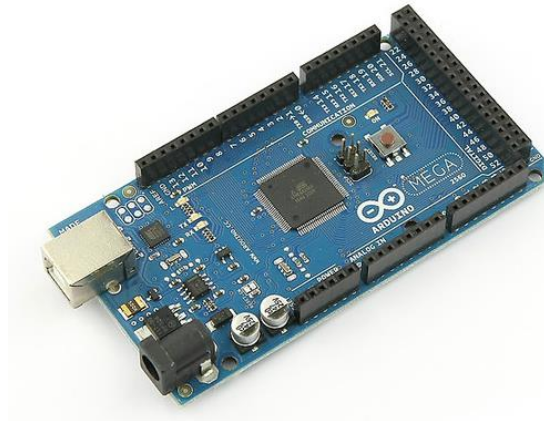
Orain lortutako ekuazioa, mikrokontrolagailuak seinalea eskuratzean exekutatu egingo da eta distantzia cm –tan emango da.

### 8.1.2. Prozesaketa

Prozesaketa egiteko, mikrokontrolagailu bat erabiliko da, kasu honetan, Arduino familiakoa. Arduino mota ezberdinak daude, esaterako: Mega, Uno, Due, YUN, Leonardo, etab. Kasu honetan, Matlab eta Arduino –ren arteko interakzioa egiten denez,

Mega –k memoria kopuru handiagoa dauka eta Matlab –ekin interakzioa egiteko garrantzitsua izango da. Horregatik Arduino Mega erabiliko da.

Kasu honetan ez du garrantzia serie porturik erabiltzen ez delako baina kontuan hartzekoa da, USB ataka erabiltzen denean txartelaren 0 serie portuarekin mapaturik doala, hori dela eta serie porturik erabiltzekotan, hobe da beste hiruretako bat erabiltzea, Arduino Mega –k 4 dituelako.



Irudia 8-4 Arduino Mega 2560

Iturria. <https://cutt.ly/ULvuCO>

Arduino Megak hainbat ezaugarri ezberdin dauzka, Arduino familiaren beste txartelekin alderatuz:

Taula 8-2 Arduino Megaren ezaugarriak

Neurriak	101,52mm x 53,3mm
Funtzionamendu tentsioa	5 V
Sarrera boltaia (gomendaturikoa)	7-12V
Sarrera boltaia (limitea)	6-20V
Sarrera irteera digitalen pin kopurua:	54 pin (15 PWM -arentzako)
Sarrera analogikoentzako pinak:	16
Sarrera irteera pinentzako korrante zuzena:	20 mA
3,3 V -ko pinaren korrante zuzena	50 mA
Flash memoria:	256 KB (8KB bootloader)
SRAM:	8 KB
EEPROM:	4 KB
Erloju abiadura:	16 MHz
Txartelan dauden Led -ak	13
Pisua	37 g

Gailu honetan programa txertatuko da eta gero bere barnean exekutatu da. Sentsorearen sarrera analogikoa eskuratuko du pin analogiko batetik eta Arduinoaren ADC –ra joango da. ADC horrek TTL teknologia erabiltzen du (0-5 V), 10 biteko erresoluzioa dauka eta 0-1023 balioen arteko neurketak egin ditzake. Posiblea da erreferentzia tentsioa 3,3 V –tara txikitzea, bereizmen handiagoa izateko seinale analogikoa digital bihurtzean. Arduino Mega –k bakarrik ADC bihurgailu bat dauka eta 16 sarrera analogiko, beraz, sarrera aukeratzeko multiplexore bat dauka. Behin seinale digitala lortuta, PID –ra joango da. Han, irteera seinalea doitu da eta horren arabera eragingailuak indar gutxiago ala gehiago izango du. Irteera PWM pin batetik aterako da eragingailura. Azaldutako prozesu hau etengabe exekutatu da, distantzia neurtu da, kontsignaren arabera PID –ak seinalea doitu da eta eragingailuak irteera aldatu da.

PID –a kontrol ingeniarietan erabiltzen den kontrolagailurik nagusia da. Kontrolagailu horrek etengabe funtzionatzen du eta berrelikaduraren bitartez etengabe aldaketak egiten ditu sistema ahalik eta gehien egonkortzeko. Hiru aldagaiz baliatzen da; P, proportzionala, I, integrala eta D, deribatiboa. PID bat sintonizatzeko, nahiz eta balioak eskuz kalkulatu, azken balioak haztamuka doitu behar dira sintonizazio fin – fina lortu arte kontrolagailua ez delako hain zehatza. Kontrolagailuaren aldagaiak azalduko dira:

- Proportzionala (P): Aldagai honek, konstante proportzionala eta errore-seinalea biderkatzen ditu. Egoera egonkorreko errorea zero baliora hurbiltzen du. Muga balio bat dago ere, lortu nahi diren balioak, konstante proportzionalako balioak baino handiagoak badira gainezkapena gertatzen da, segurtasunaren arloan garrantzi handiko gauza da. Zati proportzionalak sortu ahal dituen gehiegizko oszilazioak ez dira komenigarriak. Kontrolatzen den aldagaiaren eta bukaerako - kontrol elementuaren balioen artean erlazio lineal jarraitu bat dago. Atal honek ez du denbora kontutan hartzen, beraz ezinezkoa da errore iraunkorra konpontzea, horretarako aldagai integral eta proportzionalak daude.

$$P = u(t) = K_p * e(t)$$

- Integrala (I): Gero eta errorea txikiagoa izan doiketa azkarragoa da, baina ez da komenigarria hain azkarra izatea, sistemaren ezegonkortasuna ekar baitezake eragingailuan oszilazioak edo bibrazioak sortuz. Baliteke egoera egonkorrean erroreak zor ditzaketan perturbazioak agertzea eta kontrol proportzionalak

deuseztatu ezin izatea. Integratzailearen helburua errore hau ezabatu arte errore hori murriztea da. Bere funtzioa kontsignaren eta denbora aldagaiaren arteko ezberdintasuna integratzea da, ondoren eragin proportzionalera gehitzeko. Kontroladoreak errorearekiko proportzionala den irteera bat ematen duenez eragin geldoa duela adierazten du. Sistemari inertzia bat gehitzen zaionez, sintonizatzerako orduan kontuan izan behar da doiketa txar bat eginez gero errorea metatu eta sistema ezegonkortzen duela.

$$I = u(t) = K_i * \int_0^t e(t)dt$$

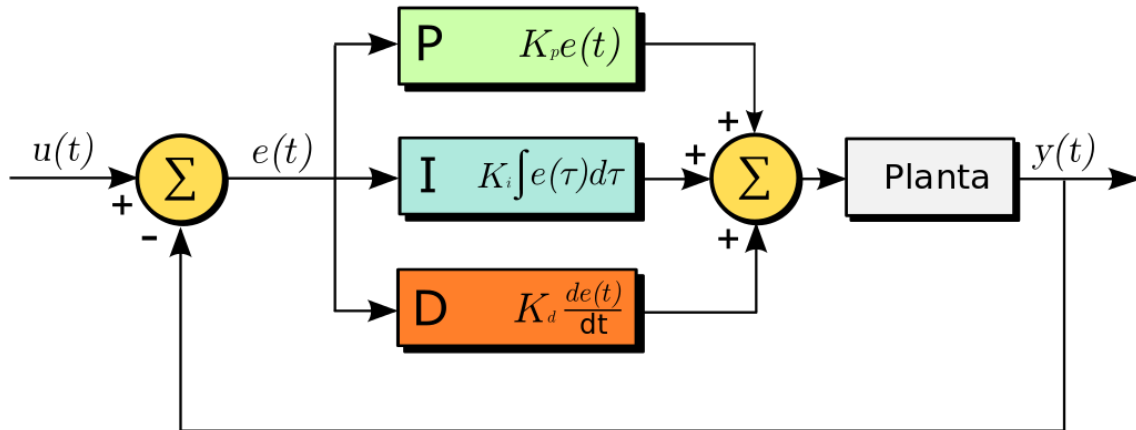
- Deribatiboa (D): Zaratarekiko oso sentikorra da. Bere funtzioa errorea aurreikusi eta tendentzia zuzentzea dira. Hau da, errorearen aldaketa abiadura baino erantzun azkarrago bat ematen du eta errorea oso handia egin baino lehen eragiten du, gainbulkadak edo oszilazioak sortzen dituzten inertziak ekiditeko asmoz. Honek errore absolutua aldatzen denean soilik eragiten du, errorea konstantea bada kontrol proportzional eta integratzaileak eragingo dute. Bere irteera aurreko errorearen eta momentuko errorearen arteko ezberdintasunarekiko proportzionala da. Bere eraginak zarata seinaleak amplifikatzen ditu prozesuan zehar eta honek eragingailua ase dezake gaizki sintonizatuta badago.

$$D = u(t) = K_d * \frac{de(t)}{dt}$$

Aurreko formulatan agertzen den deribatua, momentu zehatz batean pilotak daraman abiadura da:

$$\frac{de(t)}{dt} = \frac{e_{actual} - e_{anterior}}{T_s}$$

Hainbat kontrolagailu ezberdin prestatu daitezke, hala nola, PI, PD, P edo I motakoak. Gehien erabiltzen direnak PI –ak dira, zeren eta aldagai deribatiboa zaratarekiko oso sentikorra da eta integrala kentzea ez da komenigarria doiketa fina egiteko beharrezkoa delako.

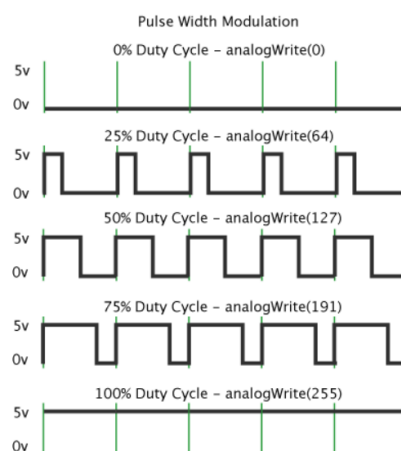


Irudia 8-5 PID kontrolagailuaren eskema eta ekuazioak

Iturria. <https://cutt.ly/kLvhlx>

Irteerako seinalea PWM seinale bat izango da zeinak lan zikloaren arabera lan egingo duen. Seinalea periodo baten barnean maila altuan dagoen denboraren portzentaiaren arabera lan egingo du, horrek, lan zikloaren balioarekin bat dator. Laburtzeko, PWM seinaleak pulsu zabalerarekin lan egiten du, beraz, timer bat erabili behar du. Zikloaren zati horretan eragingailuak lan egingo du. Seinaleak periodoa kudeatuko du 0 eta 255 balioen artean, 0 periodoaren hasiera izanik eta 255 periodoaren amaiera. PWM seinale sorketan periodoa beti maila altuan hasten da, 0 balioan, eta bakarrik adierazten da amaiera balioa, hau da, noiz emango den beheranzko saihesta.

Hurrengo irudian, seinalearen lan zikloaren adibideak agertzen dira:



Irudia 8-6 PWM seinalearen adibideak

Iturria. <https://cutt.ly/5Lvlah>

### 8.1.3. Irteerako seinalearen sorketa

Eragingailuak itxura erdi zirkularra dauka behekaldeetik, goranzko aldean, aldiz, alde batera itxura konikoa duen irteera dauka. Eragingailua kontrolatzeko PWM motako seinalea erabili behar da. Seinalea balio altuan dagoenean mugituko da, beraz, periodoa goikaldean dagoen denbora aldatuz gehiago ala gutxiago biratuko da.

Eragingailu honek PWM seinalearen bitartez lan egiten duenez bakarrik bi pin ditu, GND eta Vcc. Vcc, lan zikloarekin lan egiten du.



Irudia 8-7 COM-11270 eragingailua

Iturria. <https://cutt.ly/3LvGLX>

Eragingailuaren ezaugarri nagusiak, honako hauek dira:

Taula 8-3 Eragingailuaren ezaugarriak

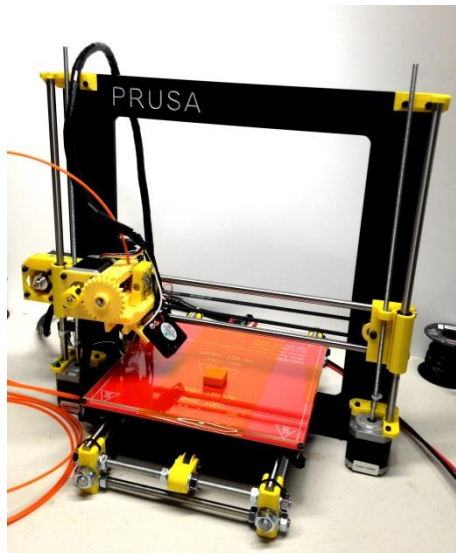
Tentsio izendatua	6-12 Vdc
Korrante izendatua	1 Amp
Aire fluxu izendatua	16 CFM
Potentzia izendatua	10 W
Irteeraren Diametroa	33 mm
Haizagailuaren Abiadura	3000 - 3500 rpm

Kasu honetan gauzarik garrantzitsuenak aire fluxuaren irteera da, konikoa denez erraza da lotura bat egitea eta ondo ahokaturik gelditzea, eragingailuaren eta tutu nagusiaren arteko aieririk galdu gabe. Horrela, funtzionamendua askoz zehatzagoa eta egonkorragoa izango da erregulazioan kanpo akatsak saihestuz. Abiadura aldetik ez dira erreboluzio gehiago behar pilota era egokian mugitzeko.

#### 8.1.4. 3D Inpresioa

3D inpresioa egiteko, Prusa i3 inprimagailua erabiliko da, FDM/FFF teknologia erabiltzen duena. Inpresio modu teknologia horrek kapen bidezko inpresioan datza, PLA plastiko mota pita baten bidez urturik ateratzen da. Material hau hurrengo arrazoiengatik aukeratu da:

- Ez da toxikoa eta Biodegradagarria da.
- 185 gradutara urtzen da.
- Solidifikatzean ez da uzkuritu.
- Abiadura handian hozten eta solidifikatzen da.



Irudia 8-8 Prusa i3 inprimagailua

Iturria. <https://cutt.ly/WLvzNs>

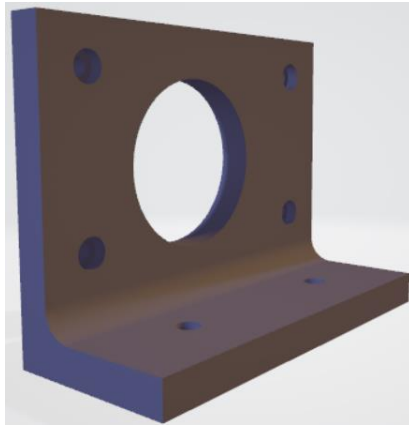
Eragingailuaren eta tutu luzearen arteko konexioa eta oinarria egiteko 3D inpresioa erabili da, horrela, eragingailua era errazean finkatuko da mahai batean. Plastikozko L itxurako 2 pieza egin dira, horrela eragingailua finkaturik gelditu da eta aire fluxuaren irteera antolaketa bertikalean. Eragingailuaren irteera konikoa eta pilota doan tutua neurri zehatzekoak direnez, konexioa egiteko pieza bat egin behar da.

Piezak egiteko, lehenengo diseinua eskuz egin da paper batean, gero Solid Edge (CAD programa) programarekin ordenagailuan piezak egin dira. Hori amaitzean, STL

formatuan esportatu dira eta amaitzeko, Cura izeneko programan piezak inpresiorako prestatu dira.

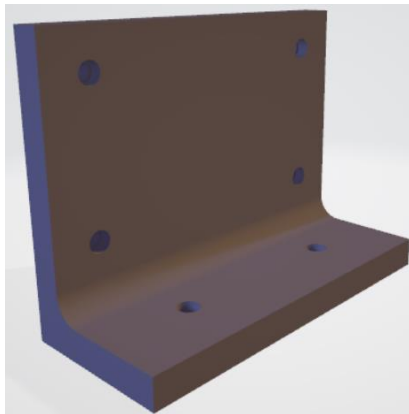
Piezen diseinuak inpresiorako prest daudenean, inprimagailuan sartzen dira eta piezen sorketa hasten da.

Hona hemen piezen 3D ereduak:



Irudia 8-9 Aire eskuratzen duen oinarriaren 3d ereduaren atala

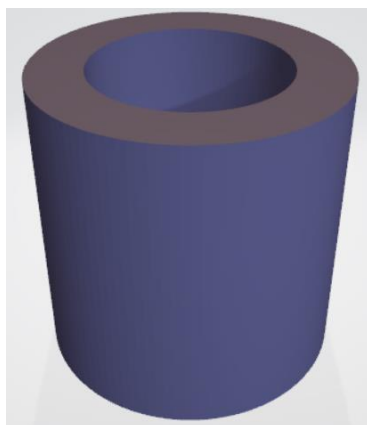
*Iturria. Berezko Elaborazioa.*



Irudia 8-10 Oinarriaren 3d ereduaren atal trinkoa

*Iturria. Berezko Elaborazioa.*



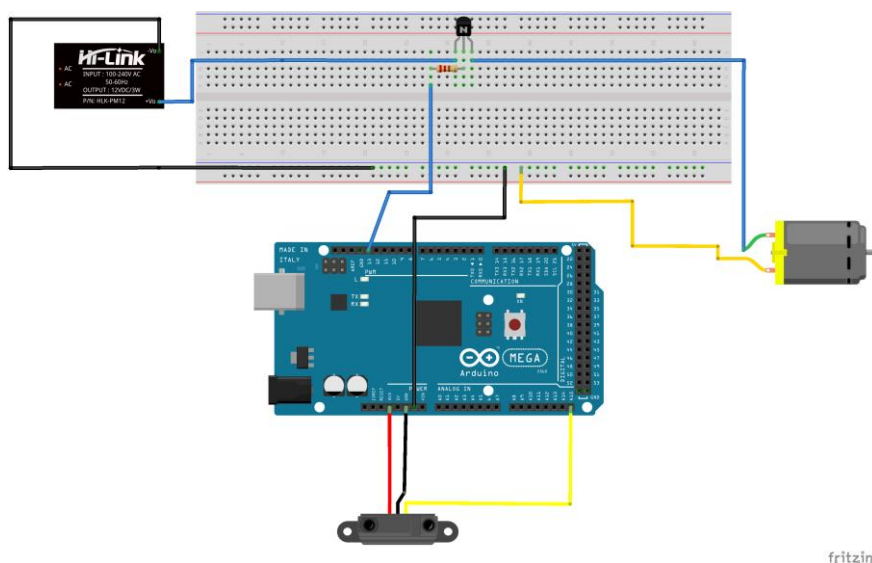


Irudia 8-11 Eragingiluaren eta pilotaren tutuaren konexioa

*Iturria. Berezko Elaborazioa.*

### 8.1.5. Konexioen Eskema

Atal honetan gailuak zehatz-mehatz nola konektaturik dauden adieraziko da. Eragingailuak 3 pin ditu, alde batetik, GND eta Vcc (3,3 V) eta beste batetik V<sub>0</sub>, Arduino -aren A15 pinera konektaturik. PWM irteera 13. Pinetik ateratzen da 330k  $\Omega$  -eko erresistore batera, gero NPN motako 2N2222 transistore baten oinarrira konektatuko da. Kolektorea 12 V -eko tentsio iturrira konektatuko da eta emisorea motorraren alde positibora. Motorraren alde negatiboa lurrera doa.



Irudia 8-12 Konexioen eskema

*Iturria. Berezko Elaborazioa.*

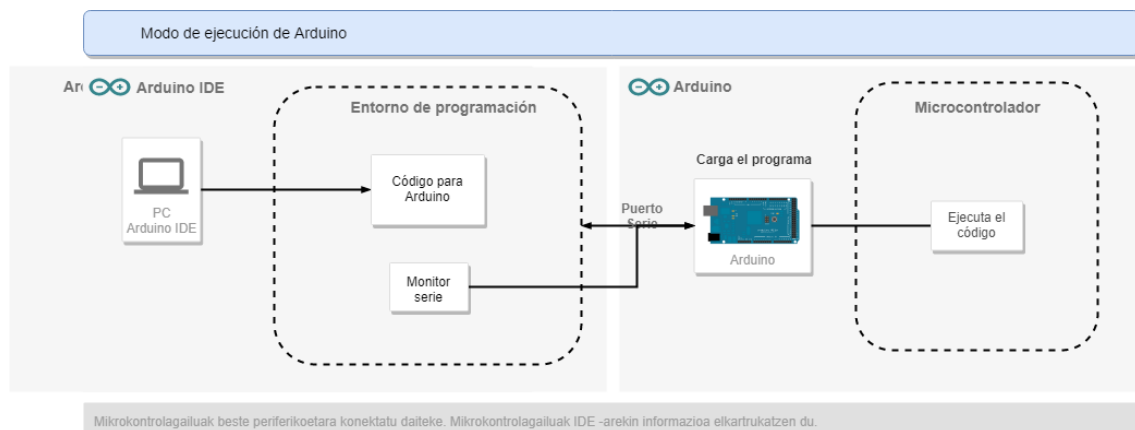
## 8.2. Software Atala

Software –ari dagokionez, proiektu hau bi atal nagusietan banatzen da, Arduino programazio ingurunea eta Matlab (Simulink) programazio ingurunea.

### 8.2.1. Arduino

Gaur egun, sistema txertatuen garapena dela eta, hainbat mikrokontrolagailu garatu egin dira, esan bezala Arduino gaur egun garrantzitsuenetakoa da. Mikrokontrolagailuez gain, Arduino bere interfaze propioa dauka, ARDUINO IDE deiturikoa. Interfaze horrek, Arduino familiaren mikrokontrolagailu guztiak programatzea ahalbidetzen du, berezko ingurune bat edukiz.

Arduino ingurunea C kodearen bitartez programatu egiten da, hori dela eta hainbat baliabide erabil daitezke, batez ere, C programazio lengoia oso ahalmen handia duelako.



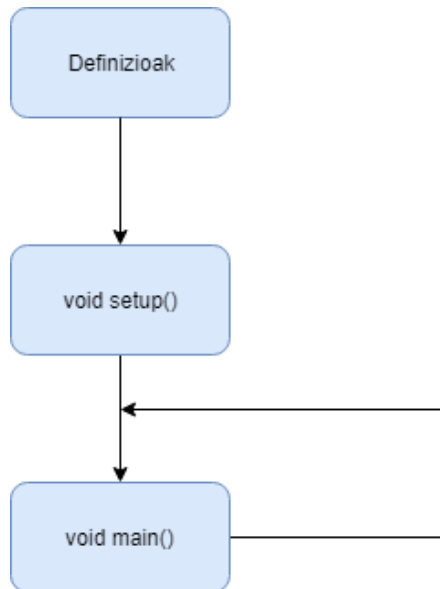
**Irudia 8-13 Mikrokontrolagailua eta Arduino IDE -aren arteko interakzioa**

*Iturria. Berezko Elaborazioa.*

Arduino IDE ingurunea mikrokontrolagailua programatzeko, orokorrean hiru pausu nagusi bereizten dira:

- Aldagaien eta liburutegien definizioa, hala nola, includeak, defineak, aldagaiak, funtzioen definizioak...

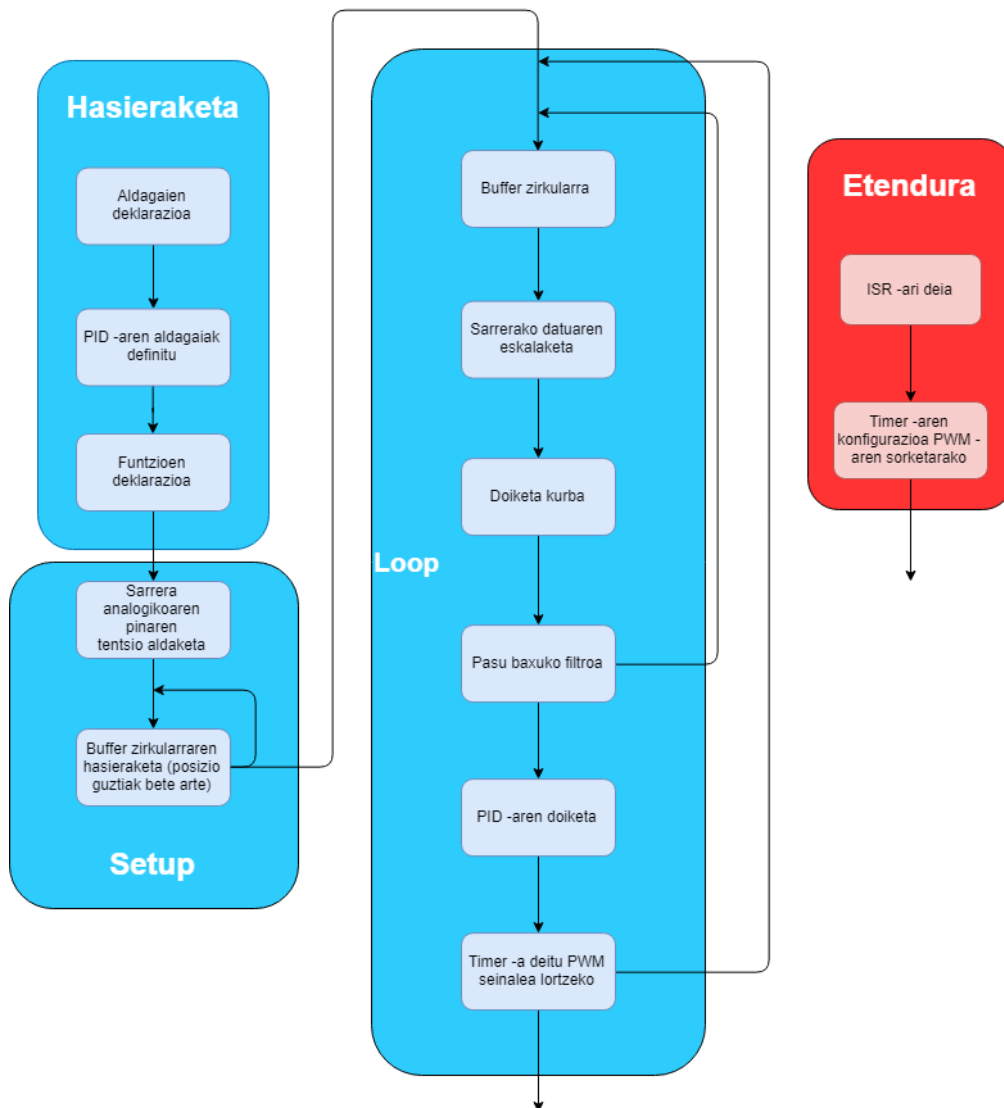
- Setup: Aldi bakar batean exekutatzen den atala. Atal honetan, programaren hasieraketa ematen da, sarrera/irteera atakak konfiguratu dira, etendurak, aldagaien hasieraketa...
- Loop: Hemen programa nagusia sartzen da eta programazioaren exekuzioa burutzen da. Etengabe exekutatzen den atala da.



**Irudia 8-14 Arduino programa baten atal nagusiak**

*Iturria. Berezko Elaborazioa.*

Pilotaren altueraren kontrola garatzeko egin den programa sortzeko, c programazioari buruzko hainbat baliabide erabili egin dira, hala nola, buffer zirkularra, sentsorearen eskala aldaketa bereizmen handiagoa edukitzeko, sentsorearen doiketa kurba...



Irudia 8-15 Arduino programaren egitura

*Iturria. Berezko Elaborazioa.*

**Buffer zirkularra:** Erabilitako sentsorearen datuak hartzerakoan, datu horiek egonkortzeko, array batean balioak sartzen dira eta arraya betetzen denean, balioen batuketa egiten da, gero balio zaharrena ateratzen da eta batuketa totalen balio horren kenketa egiten da, datu guztiak posizio bat mugitzen dira aurrera, datu berria sartzen da, datuaren batuketa egiten da eta amaitzeko batezbestekoa egiten da. Hori programaren exekuzio bakoitzean egiten da, beraz bakarrik balio bat aldatzen da exekuzio bakoitzean, horrek laguntzen du seinalearen egonkortasuna lortzera. Buffer hauetan, bi erakusle behar ditugu, bata sarrerako datuentzat eta bestea irteerarentzat.

					<b>0</b>
5	4	3	2	1	<b>15</b>
5	4	3	2		<b>14</b>
	5	4	3	2	<b>14</b>
6	5	4	3	2	<b>20</b>

Irudia 8-16 Buffer zirkularren azalpena

*Iturria. Berezko Elaborazioa.*

**Sentsorearen eskala aldaketa 5 V -tik 3,3 V -tara:** Sentsoreak 3,3 V –rekin lan egiteko nahikoa du, hori dela eta sentsorearen bereizmena handitzeko, 0–3,3 V -ko eskala erabili da. Tarte txikiagoa denez, laginen arteko tarte txikiagoa izango da, beraz bereizmena handiagoa izango da. 1024 balio egongo dira beti tarte horretan, eskala bat zein bestea erabiliz, Arduino mikrokontrolagailuaren ADC –a 10 bitekoa delako.

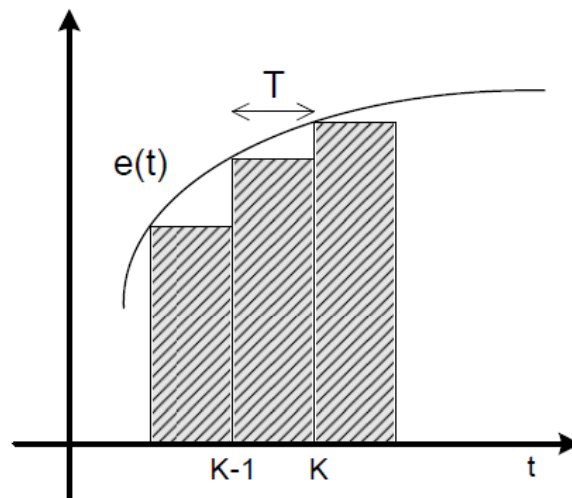
**Doiketa kurba:** Sentsorearen irteera ahalik eta gehien linealizatzeko lortutako kurba erabiliko da. Software atalean lortutakoa:

$$y = 1,2502x^6 - 16,396x^5 + 84,983x^4 - 225,46x^3 + 330,8x^2 - 272,1x + 120,72$$

**LPF:** Maiztasun baxuak pasatzen uzten dituen iragazkia da eta altuak indargabetzen ditu. Iragazki hau batez ere, zarata murrizteko erabiltzen da.

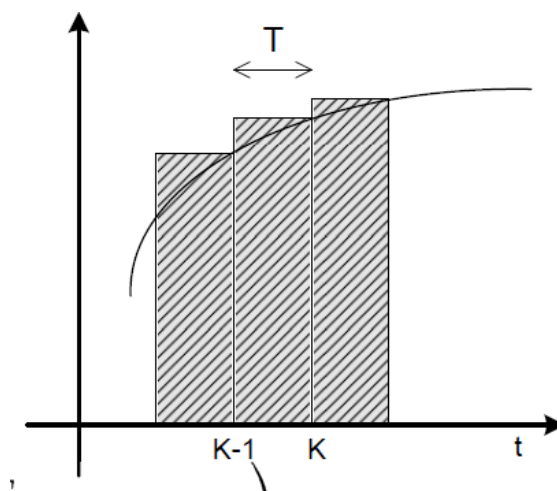
**PID:** Arduino mikrokontrolagailu diskretua da, hori dela eta PID –aren ekuazioak, zeinak sistema jarraientzako direnak, parametrizatu egin behar dira sistema diskretu batean lan egin ahal izateko. Gero PID –aren balioak doituz bere kabuz arituko da. Balioen doiketa hori egiteko, behin eta berriro konpilazioa burutu behar da, beraz, kontuan hartu behar da denboraren inbertsioa.

Ekuazioaren atal integratzailea diskretizatzeke, hurbilketa trapezoidal (edo Tustin) baten bidez egin da. Hainbat era daude, baina errore txikien ematen duen era da. Eulerren lehenengo metodoa laukizuzenen hasierako ertzen gainetik doan integrala egiten du, aldiz, Eulerren 2. Metodoa laukizuzenen amaierako ertzak hartzen ditu erreferentzia modura. Grafikoetan ikusiko den moduan lehenengo kasuan azalera gehiegi hartzen da eta bigarrean gutxiegi, hori dela eta hurbilketa trapezoidal burutuko da, baita ere bilineala deiturikoa, Eulerren bi hurbilketen arteko tarteko balio bat hartzen du eta horrela hartuko den azalera egokiagoa izango da eta errorea txikiagoa.



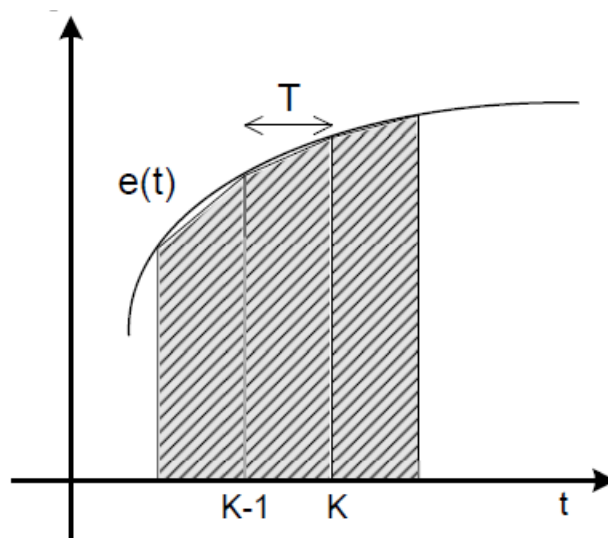
Irudia 8-17 Eulerren lehenengo hurbilketa

Iturria. <https://cutt.ly/sLvllu>



Irudia 8-18 Eulerren bigarren hurbilketa

Iturria. <https://cutt.ly/sLvllu>



Irudia 8-19 Hurbilketa Trapezoidala

Iturria. <https://cutt.ly/sLvllu>

PID –aren ekuazioan atal integratzailearen diskretizazio prozesua:

$$u(t) = k_p * e(t) + k_i * \int_t^{t+1} e(t)dt + k_d * \frac{de(t)}{dt} \left\{ \begin{array}{l} e(t) \rightarrow e[m] \\ \int_t^{t+1} e(t)dt \rightarrow \sum_m (e[m] * Ts) \\ \frac{de(t)}{dt} \rightarrow \frac{e[m] - e[m - 1]}{Ts} \end{array} \right.$$

Diskretizaturiko ekuazioa:

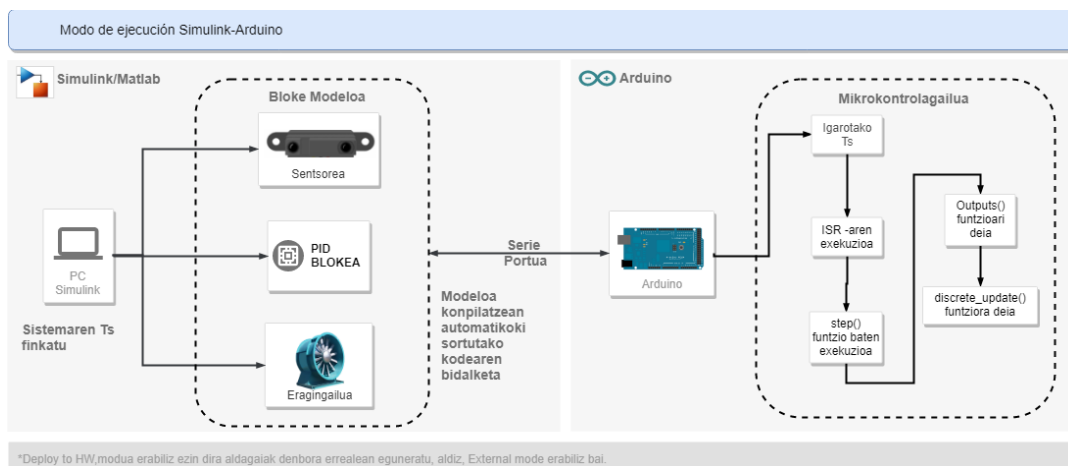
$$u(m) = k_p * e(m) + k_i * \sum_m (e[m] * Ts) + k_d * \frac{e[m] - e[m - 1]}{Ts}$$

**PWM seinalea:** Sistemaren irteera PWM seinale baten bitartez ematen da, horretarako, PWM seinale hori prestatu behar da. Seinalea prestatzeko, Timer bat erabiltzen da kasu honetan 0. timerra zeren eta PWM –ren sorkuntzarako erabiltzen den pin –a, 13. da.

## 8.2.2. Matlab – Simulink

Kasu honetan, Matlab – Simulink ingurune bisualeko programazio eremua erabiliz garatu egin da programazioa. Horretarako mikrokontrolagailuarekin elkarrekin egin beharra dago. Elkarrekin hori egiteko, Simulink –en sortutako programa mikrokontrolagailuaren barnean sartzen da eta mikrokontrolagailuaren barnean exekutatzen dira ingurune bisualean programaturiko blokeak. Exekuzioa egiterakoan bi modu ezberdin daude, alde batetik, External eta bestetik, Deploy to Hardware.

- Deploy to hardware: Programazioa simulink bidez egiten da eta exekuzioa Arduino –an egiten da. Programa Arduino -an sartzean, ordenagailua ez da ezertarako behar.
- External mode: Programazioa simulink –en egiten da eta Arduino –aren barnean exekutatzen da, baina modu honetan aldagaien aldaketa denbora errealean ematen da, baita ere emaitzak zuzenean ikuskatzea ahalbidetzen du.



Irudia 8-20 External mode

*Iturria. Berezko Elaborazioa.*

Arduino eta Matlab –en arteko interakzioa egiteko, deploy to hardware edo external mode exekutatzerakoan, Matlab –ek kodea sartzen du Arduinoan eta bi gailuen arteko elkarrekintza ematen da, elkarrekintza horren ondorioz, HTML fitxategi bat sortzen da eta han bitxikeri guztiak ageri dira. Horretarako laginketa periodo bakoitzeko, ISR bat gaituko da Ts bakoitzean. Arduino Mega erabiltzean, Timer 5 erabiltzen da 16 bitekoa



dena. Etendura hori exekutatzean, kontagailuaren baliorik handiena, Ts kontuan hartuz egingo da.

Kasu honetan 100 ms –ko laginketa denbora lortu nahi da, hau da, 0,1 s -takoa.

$$f_{bit} = \frac{16 \text{ Mhz}}{256} = 62500 \text{ Hz} \rightarrow T_{bit} = \frac{1}{f_{bit}} = 16 \mu\text{s}$$

$$T_{timer} = (2^{16} - 59286) * T_{bit} = 100 \text{ ms} \rightarrow T_{bit} = 16 \mu\text{s}$$

```

D:\Datos\Descargas\Dropbox\Dropbox\euriguen\pwmsfuncfilter2\pwmsfunc_ert_rtw>if "" == ""
("C:\PROGRA~1\MATLAB\R2018a\bin\win64\gmake" -f pwmsfunc.mk all) else ("C:\PROGRA~1\MATLAB\R2018a\bin\win64\gmake" -
f pwmsfunc.mk )
C:/ProgramData/MATLAB/SupportPackages/R2018a/3P.instrset/arduinoide.instrset/arduino-1.8.1/hardware/tools/avr/bin/avr-
g++ -std=gnu++11 -fpermissive -fno-exceptions -fno-threadsafe-statics -c -g -w -ffunction-sections -fdata-sections -MMD
-DARDUINO=10801 -MMD -MP -MF"MW_PWM.dep" -MT"MW_PWM.o" -Os -mmcu=atmega2560 -DF_CPU=16000000L -DARDUINO_AVR_MEGA2560 -
DARDUINO_ARCH_AVR -D_RUNONTARGETHARDWARE_BUILD -D_ROTH_MEGA2560_ -DMODEL=pwmsfunc -DNUMST=1 -DNCSTATES=0 -DHAVESTDIO -
DMODEL_HAS_DYNAMICALLY_LOADED_SFCNS=0 -DON_TARGET_WAIT_FOR_START=1 -DCLASSIC_INTERFACE=0 -DALLOCATIONFCN=0 -DTID01EQ=0 -
DTERMFCN=1 -DONESTEPFCN=1 -DMAT_FILE=0 -DMULTI_INSTANCE_CODE=0 -DXT_MODE=1 -DINTEGER_CODE=0 -DMT=0 -DTID01EQ=0 -DON_TARGET_WAIT_FOR_START=1 -
DXTMODE_DISABLEPRINTF -DXTMODE_DISABLETESTING -DXTMODE_DISABLE_ARGS_PROCESSING=1 -DSTACK_SIZE=64 -
D_MW_TARGET_USE_HARDWARE_RESOURCES_H_ -DRT -DMW_TIMERID=5 -DMW_PRESCALAR=256 -DMW_TIMERCOUNT=59286 -
DMW_SCHEDULERCOUNTER=1 -D_RTT_BAUDRATE_SERIAL0=115200 -D_RTT_BAUDRATE_SERIAL1=9600 -D_RTT_BAUDRATE_SERIAL2=9600 -
D_RTT_BAUDRATE_SERIAL3=9600 -D_RTT_ANALOG_REF=0 -DCLASSIC_INTERFACE=0 -DALLOCATIONFCN=0 -DTERMFCN=1 -DONESTEPFCN=1 -
DMAT_FILE=0 -DMULTI_INSTANCE_CODE=0 -DXT_MODE=1 -DINTEGER_CODE=0 -DMT=0 -DTID01EQ=0 -DON_TARGET_WAIT_FOR_START=1 -
DXTMODE_DISABLEPRINTF -DXTMODE_DISABLETESTING -DXTMODE_DISABLE_ARGS_PROCESSING=1 -DSTACK_SIZE=64 -
DRT -DMODEL=pwmsfunc -DNUMST=1 -DNCSTATES=0 -DHAVESTDIO -DMODEL_HAS_DYNAMICALLY_LOADED_SFCNS=0 -
ID:/Datos/Descargas/Dropbox/Dropbox/euriguen/pwmsfuncfilter2 -
IC:/ProgramData/MATLAB/SupportPackages/R2018a/toolbox/target/shared/svd/include -
ID:/Datos/Descargas/Dropbox/Dropbox/euriguen/pwmsfuncfilter2/pwmsfunc_ert_rtw -IC:/PROGRA~1/MATLAB/R2018a/extern/include
-IC:/PROGRA~1/MATLAB/R2018a/simulink/include -IC:/PROGRA~1/MATLAB/R2018a/rtw/c/src -
IC:/PROGRA~1/MATLAB/R2018a/rtw/c/src/ext_mode/common -IC:/PROGRA~1/MATLAB/R2018a/rtw/c/ert -

```

### Irudia 8-21 Matlab eta Arduinoaren arteko elkarrekintza

*Iturria. Berezko Elaborazioa.*

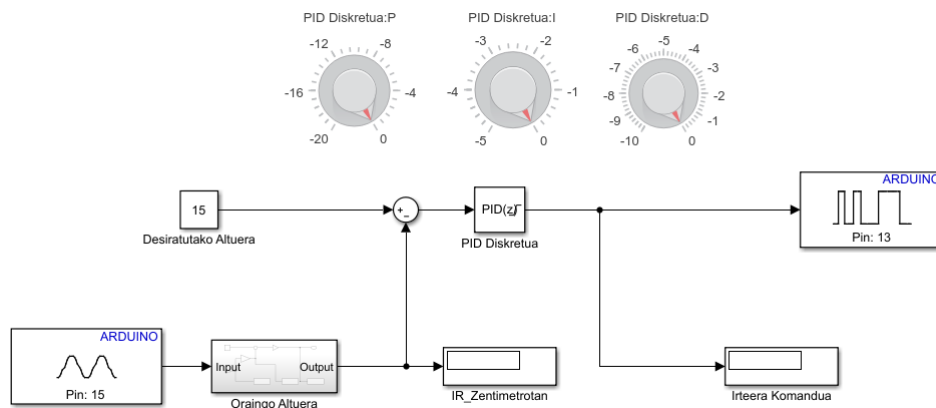
#### 8.2.2.1. Blokeen programazioa

Bi motatako programazioa egingo da blokeen bidezkoa eta s-function bidezkoa. Bi modu horiek, hiru atal nagusietan bereizten dira: datuaren eskuraketa, kontrola eta irteera. Jarraian, bi eskema nagusiak bistaratuko dira:

- Blokeen bitartez:

Simulink aldeztetik sortutako blokeak erabiliz, sistemaren eraketa nahiko erraza da, behar diren blokeak bilatzen eta kokatzen dira haien artean elkartuz, kasu honetan, Arduino -aren 15. Pinetik sarrera eskuratzen da, hurrengo azpi-blokean doitzen da eta

zuzenean PID –ra doa altueraren balioa, desiratutako altuerarekin konparatzen da eta kontrola hasten da. PID –aren sintonizazioa egitea kasu honetan oso sinplea da, balioak denbora errealean doitu daitezke denbora asko aurreztuz. Behin kontrola eginda, irteera prestatzen da PWM bloke baten bitartez.

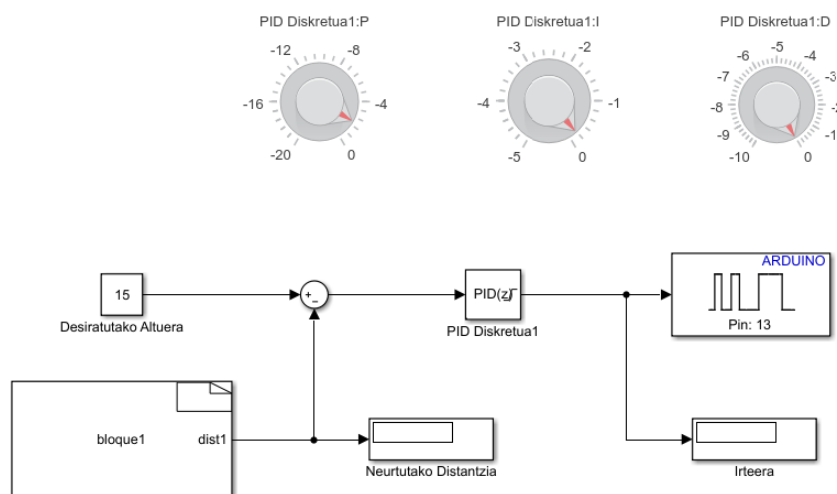


Irudia 8-22 Bloke sistemaren programa

*Iturria. Berezko Elaborazioa.*

- S-function:

Kasu honetan aldatzen den gauza bakarra seinalearen eskuraketa da, s-function bitartez egiten dena, beste atalak aurretik esan bezala berdina dira.



Irudia 8-23 S-function bloke sistema

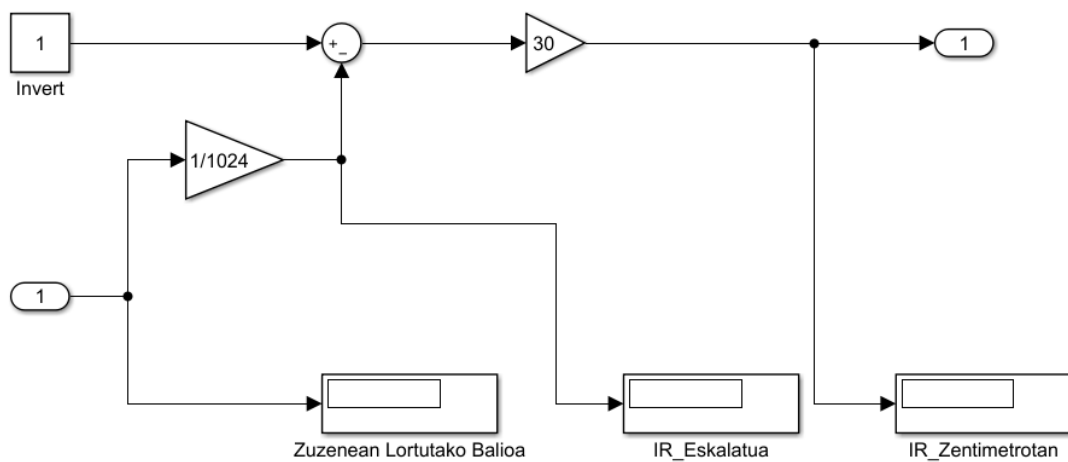
*Iturria. Berezko Elaborazioa.*

### 8.2.2.2.Datu Eskuraketa

Datua eskuratzeko bi modu erabili dira, blokeen bitartez eta s-function bitartez, seinalearen PID –aren sintonizazioa eta PWM irteera berdina da bi kasuetan. Orain seinaleen eskuraketa egiteko bi moduak azalduko dira:

- Blokeen bitartez:

Sentsoretik jasotzen den seinalea 10 bitekoa da, horregatik zati 1024 egin behar da eta gero doitu egiten da eskala zentimetroetan jartzeko. Irteeran zuzenean seinalea PID -aren barnean sar daitezke. Pausuz pausu, balioak ikusi daitezke display –etan.



Irudia 8-24 Sentsorearen seinalearen bihurketa

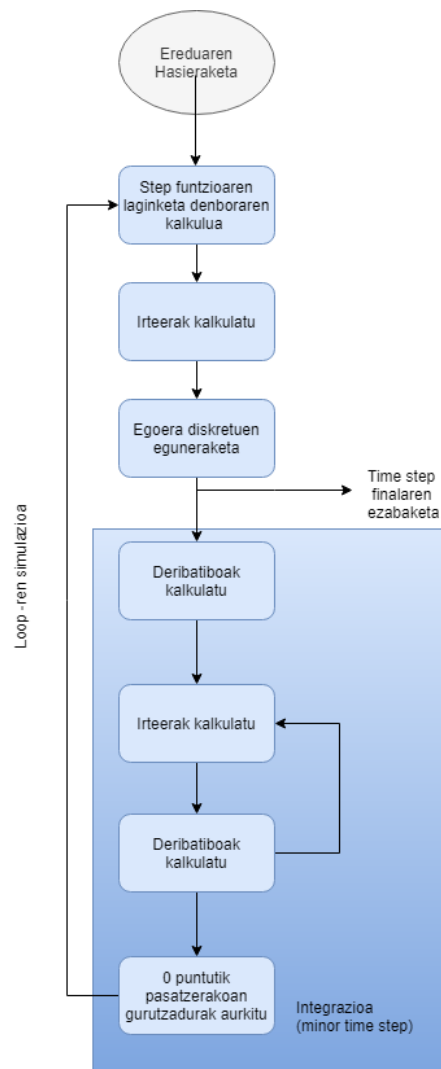
*Iturria. Berezko Elaborazioa.*

- S-function:

Blokeak erabiltzea oso aukera erakargarria da, baina Arduino atalean ikusi diren baliabideak sentsorearen seinalea eskuratzeko ezin dira erabili, beraz, ezin da erabili buffer zirkularra, pasu baxuko iragazkia, sentsorearen eskala aldaketa edo doiketa kurba. Errekurtso horiek erabili ahal izateko, s-function builder blokea dago. Bloke horrek, c programazioa simulink –en erabiltzea ahalbidetzen du. Horrela c programazioaren ahalmen konputazional guztia erabili daitezke. Bloke honen bitartez seinalearen eskuraketa askoz finagoa izango da.

Funtzionamendua, egoera makina baten bezalako da. Outputs atalean egoera exekutatu da eta Update atalean hurrengo egoerarako trantsizioa ebaluatzen da.

Update atala etengabe exekutatzen da laginketa bakoitzean. Egindako programan, Update atala doitzen da bakarrik egoera aldaketa bat emateko  $x[D]==0$  denean, zeren eta Output atalean bakarrik egoera bat definitzen da, beraz, ez da beharrezkoa egoera aldaketarik egotea.



Irudia 8-25 s-function exekuzio modu orokorra

*Iturria. Berezko Elaborazioa.*

Funtzio honek, simulink aurredefinituriko blokeen programazioa ahalbidetzeko, simulink -ek blokearen arabera badauzka txantiloiak eta han s-function -ren atalak txertatuko dira.



Irudia 8-26 s-function builder

Iturria. Berezko Elaborazioa.

Arduino -an Timer -aren etendura gaitzen denean, hau da, laginketa denbora bakoitzeko, step() funtzio bat exekutatu da eta funtzio horrek s-function -ean dagoen outputs() funtzioari dei bat egiten dio eta baita ere discrete\_update(), funtzioari.

```

80 /* Model step function */
81 void pwmsfunc_step(void)
82 {
83     real_T rtb_Sum;
84     real_T rtb_Reciprocal;
85     real_T Integrator;
86     real_T TmpSignalConversionAtFilterDiff;
87     real_T Integrator_tmp;
88
89     /* SampleTimeMath: '<S2>/TSamp' incorporates:
90      * Constant: '<S2>/N_Copy'
91      *
92      * About '<S2>/TSamp':
93      * y = u * K where K = ( w * Ts )
94      */
95     rtb_Sum = pwmsfunc_P.PIDDiskretua1_N * pwmsfunc_P.TSamp_wTEt;
96
97     /* Math: '<S2>/Reciprocal' incorporates:
98      * Constant: '<S2>/Constant'
99      * Sum: '<S2>/SumDen'
100      *
101      * About '<S2>/Reciprocal':
102      * Operator: reciprocal
103      */
104     rtb_Reciprocal = 1.0 / (pwmsfunc_P.Constant_Value + rtb_Sum);
105
106     /* SignalConversion: '<S2>/TmpSignal ConversionAtFilter Differentiator TFInport2' incorporates:
107      * Constant: '<S2>/Constant'
108      * Product: '<S2>/Divide'
109      * Sum: '<S2>/SumNum'
110      */
111     TmpSignalConversionAtFilterDiff = (rtb_Sum - pwmsfunc_P.Constant_Value) *
112     rtb_Reciprocal;
113
114     /* S-Function (bloque1): '<Root>/S-Function Builder' */
115     bloque1_Outputs_wrapper(&pwmsfunc_B.SFunctionBuilder,
116     &pwmsfunc_DW.SFunctionBuilder_DSTATE);

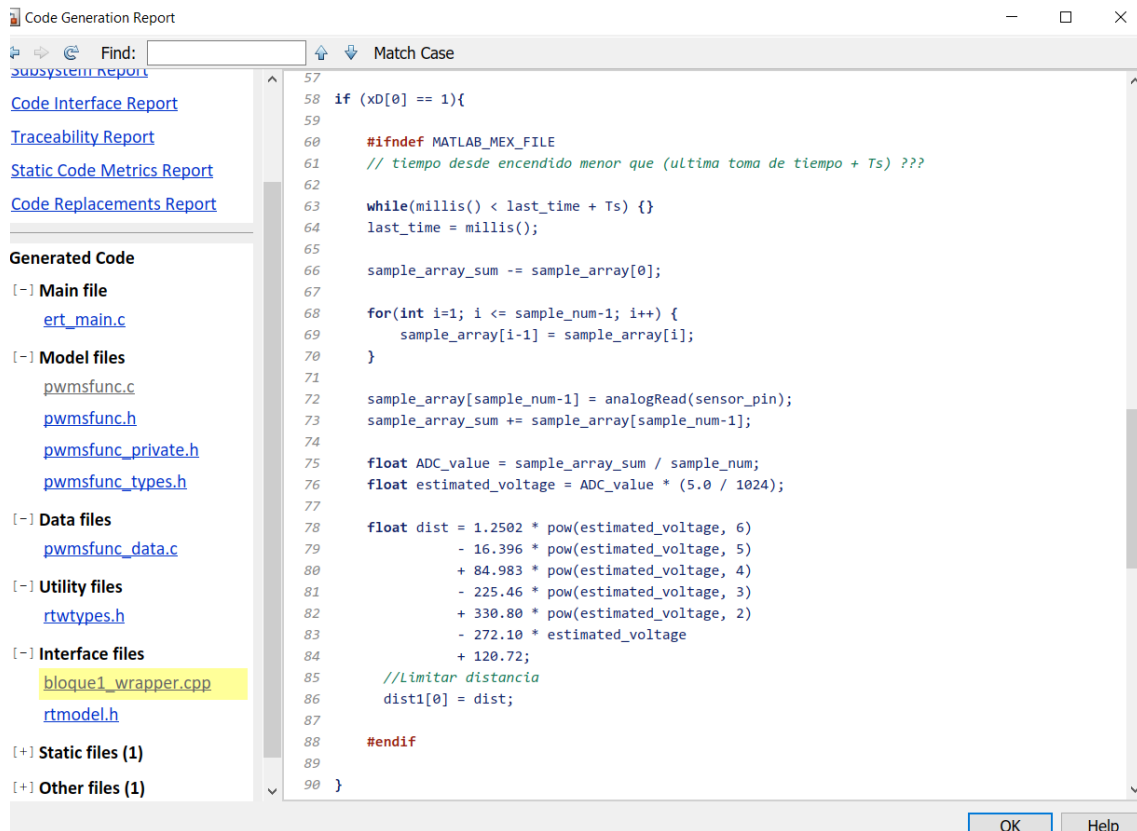
```

Irudia 8-27 Step funtzioaren sorketa

Iturria. Berezko Elaborazioa.

Ereduaren Ts –a ez badu s-function -aren Ts –arekin bat etorri, bi step() funtzio sortuko dira. Bata, ereduaren oinarri laginketa periodoarekin eta bestea, blokearen Ts –rako,

horrek kontatzen du zenbat aldiz exekutatu den timer  $T_s$ -a, zeren eta blokearen  $T_s$ -a ereduaren  $T_s$ -aren multiploa izan behar da.

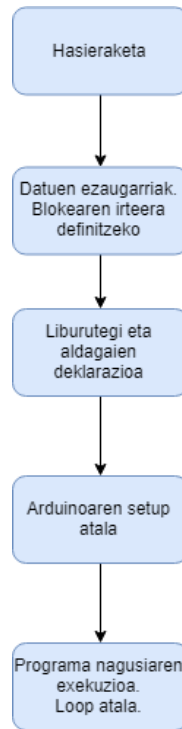


```
57
58 if (xD[0] == 1){
59
60     #ifndef MATLAB_MEX_FILE
61         // tiempo desde encendido menor que (ultima toma de tiempo + Ts) ???
62
63         while(millis() < last_time + Ts) {}
64         last_time = millis();
65
66         sample_array_sum -= sample_array[0];
67
68         for(int i=1; i <= sample_num-1; i++) {
69             sample_array[i-1] = sample_array[i];
70         }
71
72         sample_array[sample_num-1] = analogRead(sensor_pin);
73         sample_array_sum += sample_array[sample_num-1];
74
75         float ADC_value = sample_array_sum / sample_num;
76         float estimated_voltage = ADC_value * (5.0 / 1024);
77
78         float dist = 1.2502 * pow(estimated_voltage, 6)
79                   - 16.396 * pow(estimated_voltage, 5)
80                   + 84.983 * pow(estimated_voltage, 4)
81                   - 225.46 * pow(estimated_voltage, 3)
82                   + 330.80 * pow(estimated_voltage, 2)
83                   - 272.10 * estimated_voltage
84                   + 120.72;
85         //Limitar distancia
86         dist1[0] = dist;
87
88     #endif
89
90 }
```

Irudia 8-28 Kode sorketaren output funtzioa

*Iturria. Berezko Elaborazioa.*

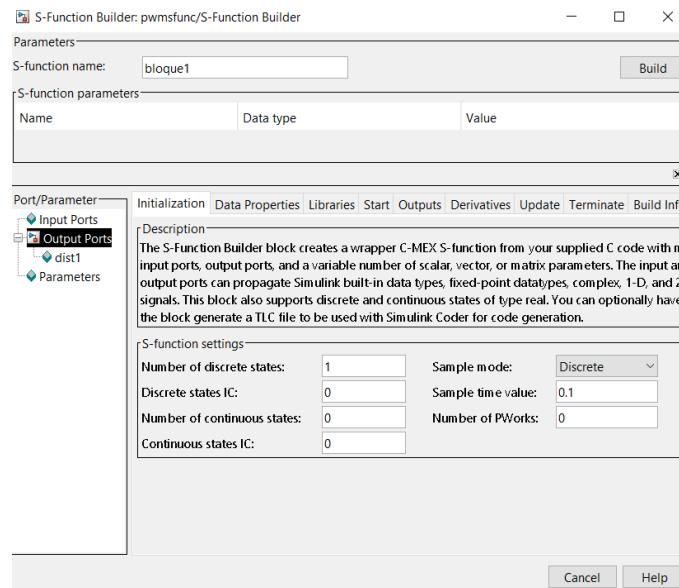
S-function builder hainbat ataletan banatzen da, hasieraketa, irteera portuen esleipena, liburutegiak, eguneraketa eta irteerak. Egitura hurrengo diagraman agertzen den banatzen da:



Irudia 8-29 s-function -ren funtzionamendu diagrama

*Iturria. Berezko Elaborazioa.*

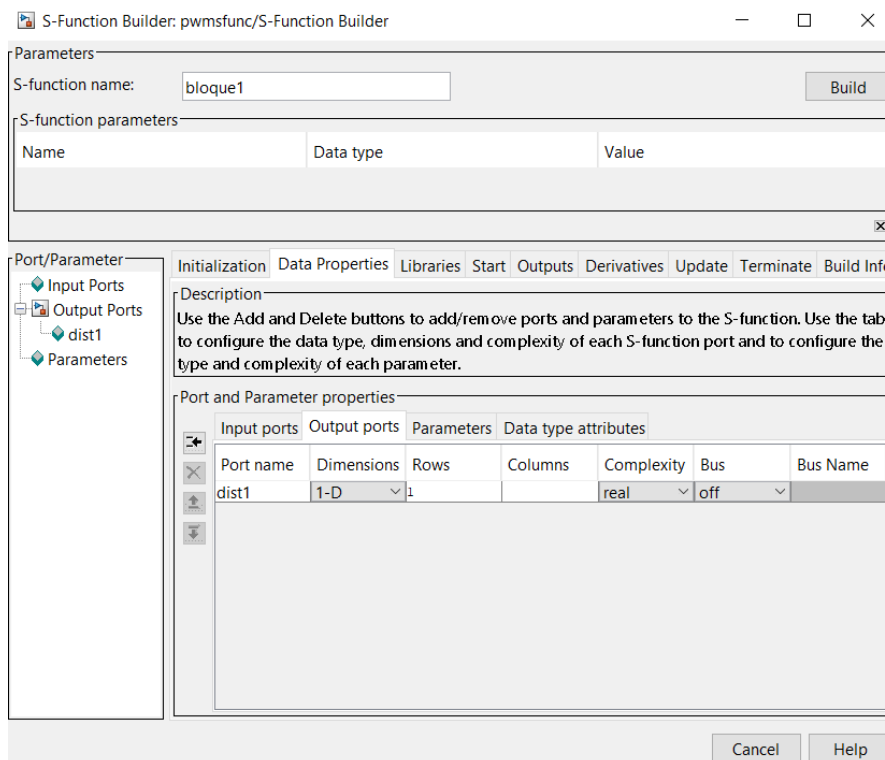
Hasieraketa, atal honetan egoera diskretuen kopurua defini daitezke, kasu honetan 1, 0 – a Arduino -aren setup atala emulatzeko eta 1 –a, Arduino -aren loop atala emulatzeko. Baita ere, sistemaren laginketa denbora hemen definitzen da, kasu honetan 100 ms.



Irudia 8-30 S-function hasieraketa atala

*Iturria. Berezko Elaborazioa.*

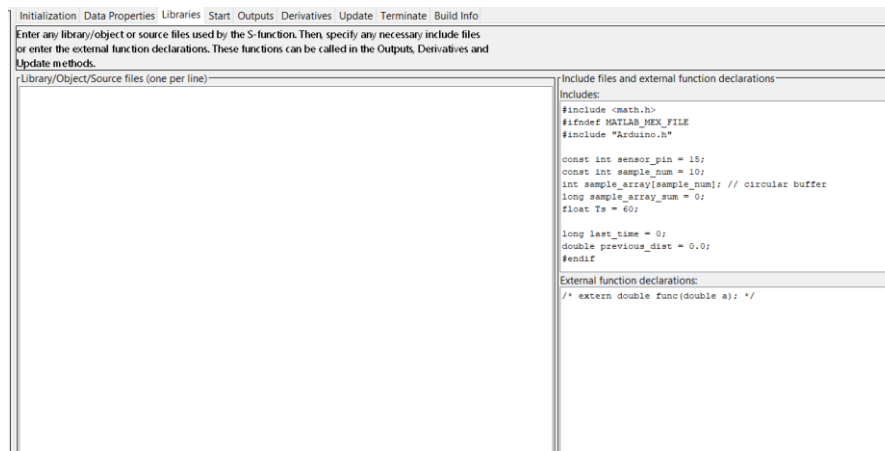
Blokearen irteerak edo sarrerak hemen definituko dira Kasu honetan dist aldagaia PID –ra sartuko da zuzenean.



Irudia 8-31 S-function irteeren atala

*Iturria. Berezko Elaborazioa.*

Liburutegien atalean, aldagaien mota, deklarazioa eta hasieraketa balioak jartzen dira, baita ere, liburutegiak, konstanteak, includeak, defineak...



Irudia 8-32 S-function liburutegiak atala

*Iturria. Berezko Elaborazioa.*



Update atalean, 0 egoera diskretua emulatzen da eta amaitzean 1 egoerara aldatzen da, horrela behin bakarrik exekutatu egingo da, hau da, Arduino -aren setup atala bezala funtzionatzen du. Kasu honetan, buffer zirkularren hasieraketa eta sentsorearen sarrerako pinaren definizioa egiten da. #ifndef eta #endif komandoen artean dagoen kodea, gutxienez aldi batean exekutatu dela bermatzen du. MATLAB\_MEX\_FILE, Matlab inguruneari adierazten dio, c kodea exekutatu dela. Komando hori #ifndef eta #endif komandoen artean egon behar da.

```

Initialization Data Properties Libraries Start Outputs Derivatives Update Terminate Build Info
Code description
This section is optional and used to update the discrete states. It is called only if the S-function has one or more discrete states. The states of the S-function are of type double and must be referenced as xD[0]...xD[n]. Input ports, output ports and parameters should be referenced using symbols specified in Data Properties. These references appear directly in the generated S-function.
/*
 * Code example
 * xD[0] = u0[0];
 */
if (xD[0] == 0) {
    #ifndef MATLAB_MEX_FILE

    pinMode(sensor_pin, INPUT);

    for(int i=0; i <= sample_num-1; i++) {
        sample_array[i] = analogRead(sensor_pin);
        sample_array_sum += sample_array[i];
        delay(60);
    }

    #endif

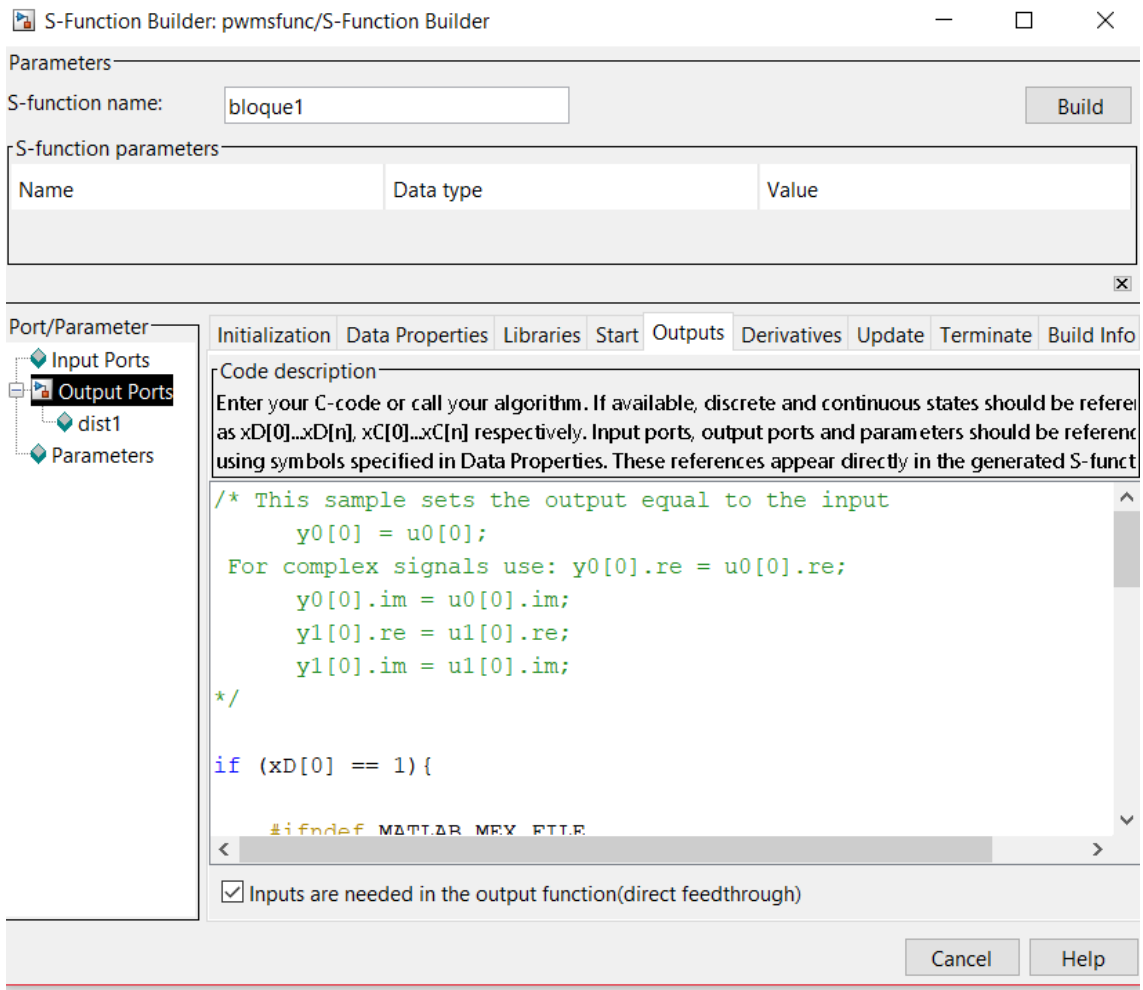
    xD[0] = 1;
}

```

Irudia 8-33 S-function eguneraketan atala

*Iturria. Berezko Elaborazioa.*

Outputs atalean, egoera diskretuaren balioa 1 bada, blokearen barnean sartuko da, hemen baita ere, #ifndef eta #endif komandoen artean c kodea exekutatzen da, MATLAB\_MEX\_FILE komandoa dago barnean c kode exekutagarria dagoela adierazteko. Simulink blokeen bitartez erabili ezin diren c kodearen baliabideak, atal honetan kokatuko dira, hala nola, buffer zirkularra, eskala aldaketa, sentsorearen doiketa kurba, pasu baxuko iragazkia. Atal hau exekutatzen da behin eta berriro, egoera diskretuaren balioa aldatzen ez bada. Arduino -aren loop atala emulatzen du.



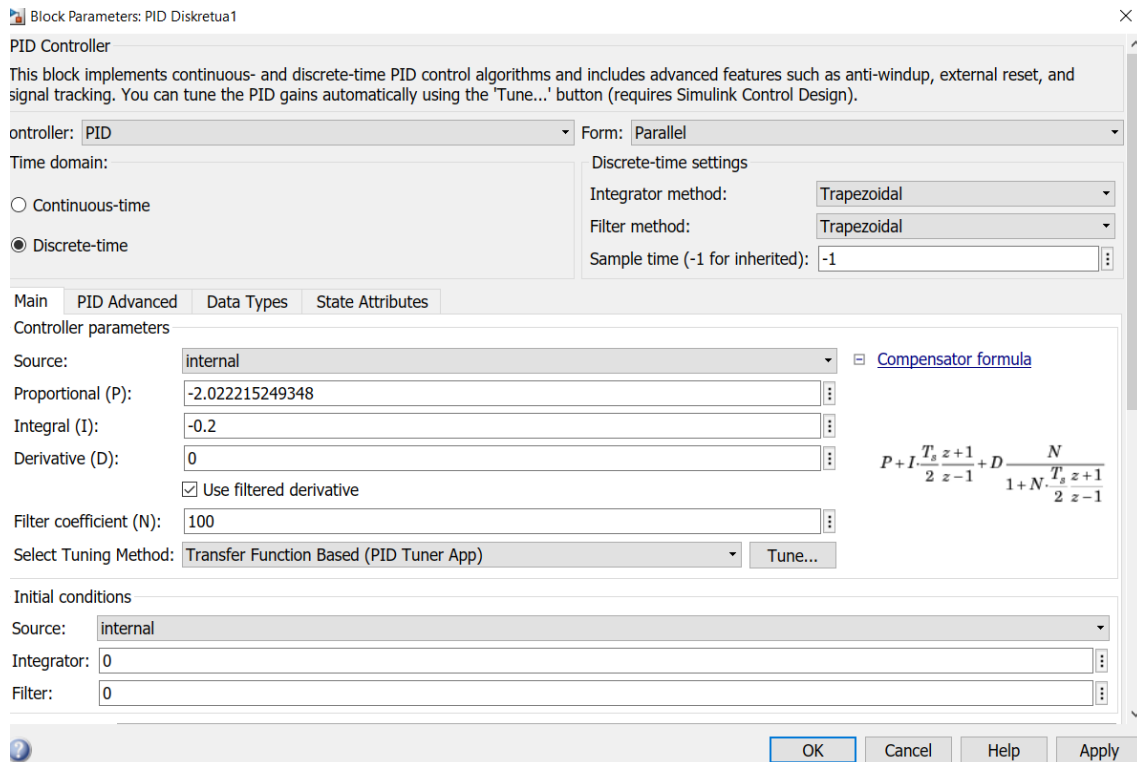
Irudia 8-34 S-function main atala

*Iturria. Berezko Elaborazioa.*

Blokea konfiguratu ostean, konpilatu egin behar da, horretarako, build botoia sakatu behar da. Konpilazioa bukatzean, Matlab –en komandoen pantaila nagusian, renc2cpp artxiboa erabili behar da, konpilatzean lortutako .c artxiboa, .cpp formatura pasatzeko Arduino -aren liburutegiak erabili ahal izateko.

### 8.2.2.3.PID –aren atala

Datua behin eskuratuta, era batetik zein bestetik, plantaren kontrola PID baten bitartez egingo da, horretarako simulink –en aurretik zehaztutako blokeak erabiliko dira, hain zuzen PID blokea eta 3 erruleta kontrolaren aldagai bakoitza kontrolatzeko, Proporzionala, Integrala eta Deribatiboa. Era horretan balioen kontrol zuzena egitea ahalbidetu egingo da. Baita ere balioak zuzenean jarri daitezke PID –aren blokean, Arduino -aren atalean azaldu den bezala, hurbilketa Trapezoidala erabiliko da.

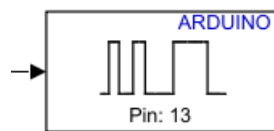


Irudia 8-35 PID blokearen xehetasunak

*Iturria. Berezko Elaborazioa.*

#### 8.2.2.4.PWM –aren sorketa

Doiketa egin ezker, sistemaren irteera seinalea atera behar da, horretarako PWM irteerari dagokion aurre zehaztutako blokea erabiliko da, bloke horren irteera 0-255 balioen artean simulink hasieratik mugatzen du. Barnetik 0. Timer -ra erabiltzen da PWM seinalea sortzeko.



Irudia 8-36 PWM irteera

*Iturria. Berezko Elaborazioa.*

## 9. PLANGINTZA

Edozein motako proiektu bat egiterakoan alde zuzenetik, zer egingo den eta nola egingo den planifikatu egin behar da atalez atal eta denbora epe batzuk jarri behar dira atal bakoitza egiteko, horrela proiektua egiterakoan, prozesuaren kontrol zehatz bat edukitzea lortuko da, bideragarritasuna lagunduz.

Alde batetik, proiektuaren faseak adieraziko dira taulen bitartez eta beste batetik, grafikoki adieraziko da plangintza, Gantt diagrama batez baliaturik.

### 9.1. Proiektuaren partaideak

Lan hau garatuko duten partaideak, hauek izango dira:

- **Zuzendaria:** Oskar Casquero Oyarzabal.
- **Ingeniari junior:** Ekain Uriguen Fernandez.
- **Kudeatzailea:** Henar Brenlla Garcia.
- **Laborategiko teknikaria:** Cesar Perez Barrio.

### 9.2. Proiektuaren paketeak

Proiektua hainbat paketetan banatuko da, hala nola, gestioa, prestakuntza, formakuntza, frogapena eta dokumentazioa. Atal bakoitzak bere garrantzia dauka eta normalean ezinezkoa da ordena aldatzea. Pakete bakoitzak bere eginkizunak dauzka, prestakuntza, formakuntza, frogapena eta dokumentazioa prozesuetan banatzen da, gestioa aldez, proiektuko pakete guztiak gainbegiratzen du.

#### 9.2.1. Gestioa

- **Deskribapena:** Lan pakete honetan, lan osoaren koordinazioa burutuko da, beraien arteko korrelazioa bermatuz.
- **Arduraduna:** Oskar Casquero Oyarzabal.

- **Parte hartzaileak:** Oskar Casquero Oyarzabal, Cesar Perez Barrio, Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 60 egun.

### 9.2.2. Prestakuntza

- **Deskribapena:** Atal honetan informazioaren bilaketa eta prestaketa egingo da.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Oskar Casquero Oyarzabal, Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 10 egun.

Lan Eskakizuna 2.1:

- **Deskribapena:** Informazioaren bilaketa.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez.
- **Iraupena:** 4 egun.

Lan Eskakizuna 2.2:

- **Deskribapena:** Informazioaren prestaketa.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 3 egun.

Lan Eskakizuna 2.3:

- **Deskribapena:** Aurretiko plangintza.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 2 egun.

Lan Eskakizuna 2.4:

- **Deskribapena:** Zuzendariarekin elkarketa.
- **Arduraduna:** Oskar Casquero Oyarzabal.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Oskar Casquero Oyarzabal.
- **Iraupena:** 1 egun.

### 9.2.3. Formakuntza

- **Deskribapena:** Hemen erabiliko diren gailuen eta makinen frogak eta ikasketak egingo dira
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Oskar Casquero Oyarzabal, Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 15 egun.

Lan Eskakizuna 3.1:

- **Deskribapena:** Arduinoarekin trebetasunak lortu, programak eta mikrokontrolagailuarekin.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 3 egun.

Lan Eskakizuna 3.2:

- **Deskribapena:** Matlab –ekin trebetasunak lortu.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 4 egun.

Lan Eskakizuna 3.3:

- **Deskribapena:** Sentsoreakin gaitasunak lortu, doiketa kurba egin.
- **Arduraduna:** Ekain Uriguen Fernandez.

- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 2 egun.

Lan Eskakizuna 3.4:

- **Deskribapena:** Eragingailuarekin frogak, PWM seinalearen ikasketak.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 2 egun.

Lan Eskakizuna 3.5:

- **Deskribapena:** Sentsorea eta eragingailuarekin elkarrekin frogak egitea.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 3 egun.

Lan Eskakizuna 3.6:

- **Deskribapena:** Tutorearekin batzarra.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Oskar Casquero Oyarzabal, Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 1 egun.

#### 9.2.4. Frogapena

- **Deskribapena:** Muntaketa eta proiektuaren finkapena.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 20 egun.

Lan Eskakizuna 4.1:

- **Deskribapena:** Maketaren muntaketa.
- **Arduraduna:** Cesar Perez Barrio.

- **Parte hartzaileak:** Cesar Perez Barrio, Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 3 egun.

Lan Eskakizuna 4.2:

- **Deskribapena:** Arduino programa egin.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 4 egun.

Lan Eskakizuna 4.3:

- **Deskribapena:** Arduino programan, PID –aren doiketa.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 5 egun.

Lan Eskakizuna 4.4:

- **Deskribapena:** Matlab –en modeloak egin.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 4 egun.

Lan Eskakizuna 4.5:

- **Deskribapena:** Matlabeko modeloen PID -aren doitu.
- **Arduraduna:** Oskar Casquero Oyarzabal.
- **Parte hartzaileak:** Ekain Uriguen Fernandez eta Henar Brenlla.
- **Iraupena:** 3 egun.

Lan Eskakizuna 4.6:

- **Deskribapena:** Tutorearekin batzarra.
- **Arduraduna:** Oskar Casquero Oyarzabal.
- **Parte hartzaileak:** Oscar Casquero Oyarzabal, Ekain Uriguen Fernandez eta Henar Brenlla.



- **Iraupena:** 1 egun.

#### 9.2.5. Dokumentazioa

- **Deskribapena:** Dokumentazioa.
- **Arduraduna:** Oskar Casquero Oyarzabal.
- **Parte hartzaileak:** Oskar Casquero Oyarzabal eta Ekain Uriguen Fernandez.
- **Iraupena:** 15 egun.

Lan Eskakizuna 5.1:

- **Deskribapena:** Dokumentazioa
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez.
- **Iraupena:** 12 egun.

Lan Eskakizuna 5.2:

- **Deskribapena:** Dokumentazioaren berrikuspena.
- **Arduraduna:** Ekain Uriguen Fernandez.
- **Parte hartzaileak:** Ekain Uriguen Fernandez.
- **Iraupena:** 2 egun.

Lan Eskakizuna 5.3:

- **Deskribapena:** Tutorearekin batzarra.
- **Arduraduna:** Oskar Casquero Oyarzabal.
- **Parte hartzaileak:** Oskar Casquero Oyarzabal eta Ekain Uriguen.
- **Iraupena:** 1 egun.

Aurretik azaldutako lan pakete eta eginkizunez aparte, hainbat mugarri definitu egin dira, proiektuaren garapena denboran ebaluatzeko. Proposatutako mugarriak hurrengoak izan dira:

- Lan taldearen osaketa.
- Informazioa bilatuta eta hasierako planifikazioa.

- Gailuen frogaketa eta programen sorketa.
- PID –aren doiketa amaituta Matlab eta Arduinon
- Dokumentazioa idatzirik eta entregaturik.

### 9.3. Gantt Diagrama

Orain Gantt diagraman aurreko ataletan azaldutako lan paketeak, eginkizunak eta mugarrak era grafiko batean adieraziko dira. Proiektua, 2018 ko Urrian hasi egin da eta 2019 ko Uztailan amaituko da. Hilabete hauetan, lan karga ez da berdina izan, zeren eta graduko ikasgaiak burutu egin dira aldi berean. Hori dela eta, Abenduan eta Urtarrilean ia ez da ezer aurreratu graduko azterketak direla eta, aldi berean Otsailean eta Martxoan karga handitu egin da. Maiatzean eta Ekainean berriro karga gutxitu da azterketak direla eta Uztailan lan karga handitu egin da, azterketen bukaera eta lana entregatzearen epeak direla eta.

Planifikazioa egunetan egin da, alde batetik 4 orduko 42 egun Urritik Ekainera arte eta beste batetik, 9 orduko 18 egun Uztailan, horrela 330 ordu guztira aterako dira.

Taula 9-1 Lan paketeen laburpena

LP Zk	Lan Paketea	Hasiera Hilabetea	Amaitze Hilabetea	Entregagarria
LP1	Gestioa	1	10	Lanaren Jarraipena
LP2	Prestakuntza	1	2	D2.1-D2.4
LP3	Formakuntza	2	5	D3.1-D3.6
LP4	Frogapena	5	10	D4.1-D4.6
LP5	Dokumentazioa	4	10	D5.1-D5.3

Taula 9-2 Lan eskakizunen garrantzia

	Garrantzi Altua
	Garrantzi Ertaina
	Garrantzi Baxua

Taula 9-3 Gantt Diagrama

	2018			2019						
	Urria	Azaroa	Abendua	Urtarrila	Otsaila	Martxoa	Apirila	Maiatza	Ekaina	Uztaila
LP1										
LP2										
Eginkizuna 2.1										
Eginkizuna 2.2										
Eginkizuna 2.3										
Eginkizuna 2.4										
LP3										
Eginkizuna 3.1										
Eginkizuna 3.2										
Eginkizuna 3.3										
Eginkizuna 3.4										
Eginkizuna 3.5										
Eginkizuna 3.6										
LP4										
Eginkizuna 4.1										
Eginkizuna 4.2										
Eginkizuna 4.3										
Eginkizuna 4.4										
Eginkizuna 4.5										
Eginkizuna 4.6										
LP5										
Eginkizuna 5.1										
Eginkizuna 5.2										
Eginkizuna 5.3										

## 10. AURREKONTUA

Aurrekontuan, proiektu osoa garatzeko kostu ekonomikoak aztertuko dira. Hori burutzeko, hainbat alderdi kontuan hartuko dira. Horretarako aurrekontua bi ataletan bananduko da, alde batetik, garapen kontzeptualaren aurrekontua eta bestetik, proiektuaren ezarpenerako aurrekontua.

Garapen kontzeptualaren aurrekontuan, proiektuaren nondik norakoak azalduko dira, funtzionamendu eta prozedura guztiak finkatuz, materialen aukeraketa, legediak... proiektua era kontzeptualean garatuz.

Taula 10-1 Garapen Kontzeptualaren Aurrekontua

Kontzeptua	Unitateak	Unitate kopuruak	Kostu Unitarioa	Kostua	Total
Barne Orduak					9.795,00 €
Proiektuaren Zuzendaria	h	17	35,00 €	595,00 €	6.500,00 € 2.700,00 €
Ingeniari Junior	h	260	25,00 €		
Kudeatzailea	h	135	20,00 €		
Amortizazioak					103,00 €
Ordenagailua	h	412	0,25 €	103,00 €	
Kostu Zuzenak					9.898,00 €
Kostu Ez-zuzenak	10%				989,80 €
Azpi totala					10.887,80 €
Ezusteak	10%				1.088,78 €
<b>GUZTIRA</b>					<b>11.976,58 €</b>

Proiektuaren ezarpenerako aurrekontuan, proiektua exekutatzeko behar diren baliabide ekonomikoak azalduko dira. Aurretik egin den garapen kontzeptualetik abiatuz, proiektua funtzionamenduan jartzeko baliabide guztiak kontuan hartuko dira atal honetan, hala nola, muntaia egiteko parte hartzen duten langileen soldatak, makinaren amortizazioak, erabiliko diren osagaiak...

Taula 10-2 Proiektuaren Ezarpen Aurrekontua

Kontzeptua	Unitatea	Unitate kopuruak	Kostu Unitarioa	Kostua	Total
Barne Orduak					3.050,00 €
Proiektuaren Zuzendaria	h	4	35,00 €	140,00 €	1.750,00 €
Ingeniari Junior	h	70	25,00 €	€	
Kudeatzailea	h	40	20,00 €	800,00 €	
Laborategi Teknikaria	h	12	30,00 €	360,00 €	
Amortizazioak					
Ordenagailua	h	126	0,25 €	31,50 €	73,61 €
3D inprimagailua	h	8	0,30 €	2,40 €	
Matlab	h	70	0,15 €	10,50 €	
Kostuak					73,61 €
GP2Y0A21YK sentsorea		1	9,50 €	9,50 €	3.168,01 €
Eragingailua		1	5,30 €	5,30 €	
Arduino Mega 2560		1	28,00 €	28,00 €	
2N2222 transistorea		1	0,45 €	0,45 €	
330kΩ erresistentzia		1	0,12 €	0,12 €	
Kableak	Pack 65u	1	1,19 €	1,19 €	
Inpresiorako plastikoa		1	18,27 €	18,27 €	
Ping-Pong pilota	Pack 6u	1	10,78 €	10,78 €	
Kostu Zuzenak					
Kostu Ez-zuzenak	10%				316,80 €
Azpi totala					3.484,81 €
Ezusteak	10%				348,48 €
GUZTIRA					3.833,29 €

Proiektua aurrera eramateko, garapen kontzeptualaren eta ezarpenaren kostuak batuz, proiektuaren kostu totala lortuko da, proiektatzen denetik, fisikoki muntaia amaitzen denera arte.

Taula 10-3 Kostu Totala

Kontzeptua	
Garapen Kontzeptuala	11.976,58 €
Proiektuaren Ezarpena	3.833,29 €
GUZTIRA	15.809,87 €

## 11. ONDORIOAK

Atal honetan, proiektua garatu ostean aztertu diren xehetasunak azalduko dira, lan luze batean hainbat gertakari agertzen dira, horren ondorioz hemen azalduko dira proiektu osoa aztertu eta gero lortu diren ondorioak.

Arduino bitartez lan egiterakoan, erabili behar den mikrokontrolagailuaren ezaugarriak ez direla gaitasun handikoak izan behar ikusten da, alde batetik, soilik timer 1 erabiltzen delako eta beste batetik programa oso trinkoa denez, memoria gutxiago behar delako. C programazio lengoiaia oinarritzkoena da baina aldi berean, oso gaitasun handiko lengoiaia da horregatik programazioa asko trinkotu daiteke.

Matlab bidez programatzeak ez da hain zaila ingurune bisuala dela eta, beraz ez da hain abstrakzio handia izan behar, C lengoiaian gertatzen ez den bezala, ingurune horretan lan egiteko. Mikrokontrolagailuak hainbat gaitasun behar ditu Matlab –ekin interakzioa egin ahal izateko, Matlab –ek sortzen duen kodea ez da hain eraginkorra, beraz, baita ere memoria gehiago behar da interakzio hori era egokian burutu ahal izateko.

Matlab –en sensorearen irakurketa egiteko, bi modu aztertu dira, s-function alde batetik eta beste batetik Matlab –ek dakartzan blokeak. Ikusi den ondorio nagusia hauxe izan da, c lengoiaia eta Matlab –eko blokeak aldi berean erabiltzea hoberena dela, horrela c lengoiairekin gaitasun handiarekin sensorearen neurketa hobeak egingo dira eta PID –aren doiketa askoz hobeagoa izango da.

PID –aren doiketa zeregin neketsua da doiketa on bat lortzeko hainbat balio ezberdin frogatu behar direlako, Arduino -aren bidez lan egitean aldaketa bakoitza egiterakoan, berriro konpilatu behar da programa osoa horrek dakartzan denbora galketarekin. Ondorioz Matlab bitartez lan egitean Arduino -arekin interakzioa egin ahal da zuzenean external modua erabiliz, horrela Arduino ingurunean konpilazio bakoitza egitean galtzen den denbora aurrezten da, beraz, horrek prozesua asko errazten eta azkartzen du.

Amaitzeko ondorioztatu daiteke Matlab erabiltzea s-function erabiliz hoberena dela. s-function erabiltzean sensorearen datuaren eskuraketa askoz zehatzagoa eta azkarra da, gero PID –aren doiketa askoz azkarragoa da balioen aldaketak zuzenean egiten direlako, beraz, doiketa askoz denbora gutxiago erabiliz lortzen da.

## 12. BIBLIOGRAFIA

### GP2Y0A21YK distantzia sentsorea

- [1] Ezaugarriak <https://tienda.bricogeek.com/home/261-sensor-de-proximidad-sharp-gp2y0a21yk.html> (Eskuragarri: 2019-07-05)
- [2] Datasheet <http://www.sparkfun.com/datasheets/Components/GP2Y0A21YK.pdf> (Eskuragarri: 2019-07-05)

### Arduino Mega

- [3] Ezaugarriak <https://store.arduino.cc/mega-2560-r3> (Eskuragarri: 2019-07-05)
- [4] Irudia <https://store.arduino.cc/mega-2560-r3> (Eskuragarri: 2019-07-05)
- [5] Datasheet [http://www.atmel.com/dyn/resources/prod\\_documents/doc2549.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2549.pdf) (Eskuragarri: 2019-07-05)

### Eragingailua

- [6] Irudia <https://www.sparkfun.com/products/11270> (Eskuragarri: 2019-07-05)
- [7] Ezaugarriak <https://www.sparkfun.com/products/11270> (Eskuragarri: 2019-07-05)

### PWM

- [8] <https://www.arduino.cc/en/tutorial/PWM> (Eskuragarri: 2019-07-05)

### Eragingailuaren konexioak

- [9] <https://www.prometec.net/s4a-transistores-motor/> (Eskuragarri: 2019-07-05)

### Diskretizazioa

- [10] [http://www.control-class.com/Tema\\_6/Slides/Tema\\_6\\_Disen%C3%B3\\_Controladores.pdf](http://www.control-class.com/Tema_6/Slides/Tema_6_Disen%C3%B3_Controladores.pdf) (Eskuragarri: 2019-07-05)

### Arduino Timer -rak

- [11] <https://forum.arduino.cc/index.php?topic=410287.0> (Eskuragarri: 2019-07-05)

### s-function

[12] <https://www.mathworks.com/help/simulink/slref/sfunction.html> (Eskuragarri: 2019-07-05)

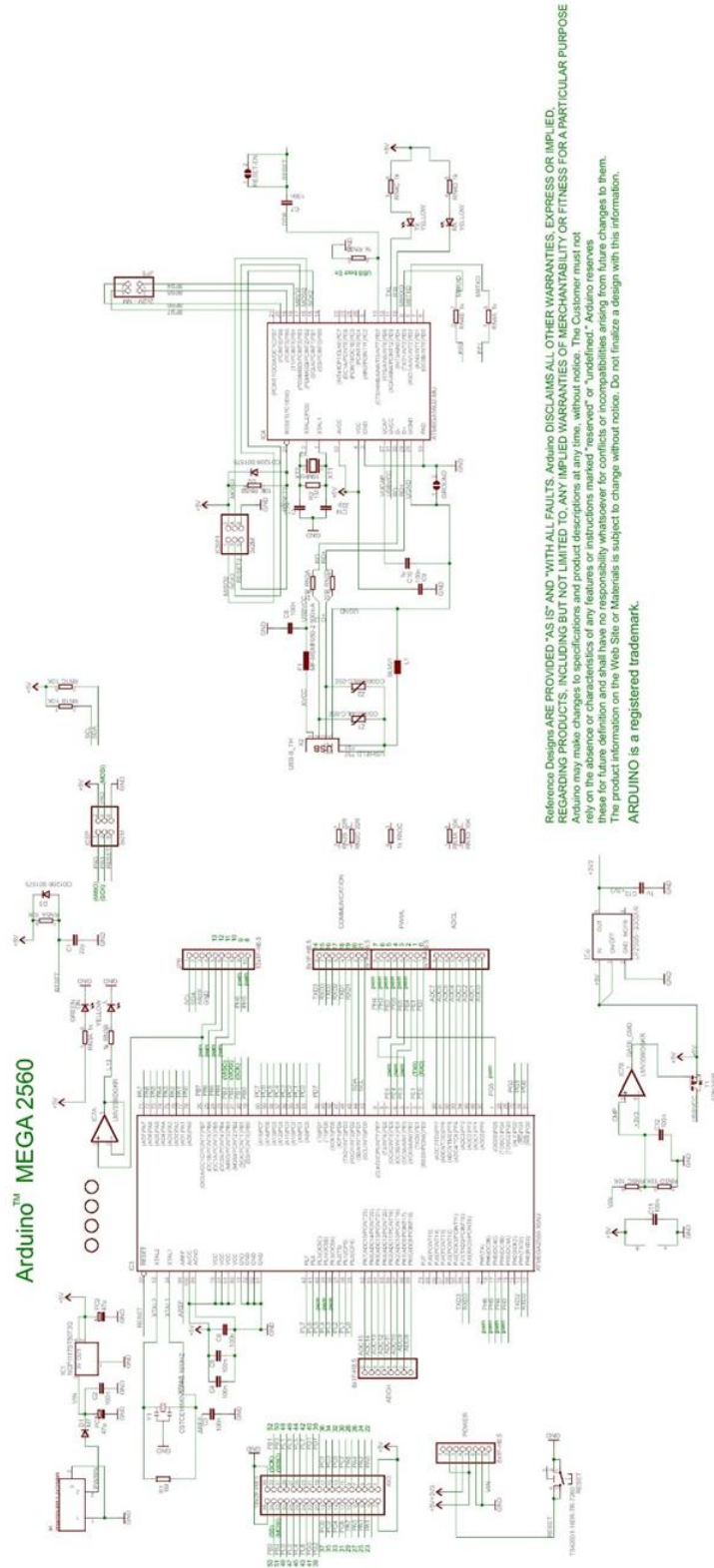
## **PID**

[13] [https://es.wikipedia.org/wiki/Controlador\\_PID](https://es.wikipedia.org/wiki/Controlador_PID) (Eskuragarri: 2019-07-05)



# ERANSKINAK 1: Arduino -aren barne eskema

Arduino mikrokontrolagailuak daraman barne konexioak:



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. ARDUINO DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "unlabeled". Arduino reserves the right to make changes to specifications and product descriptions without notice. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

## ERANSKINAK 2: Kodea eta Parametroak

Atal honetan, proiektuan erabili diren programazio kodeak atxikituko dira. Horretarako Github biltegian gorde dira. Baita ere, eredurako PID –a sintonizatzeko erabili diren balioak, han gordeko dira.

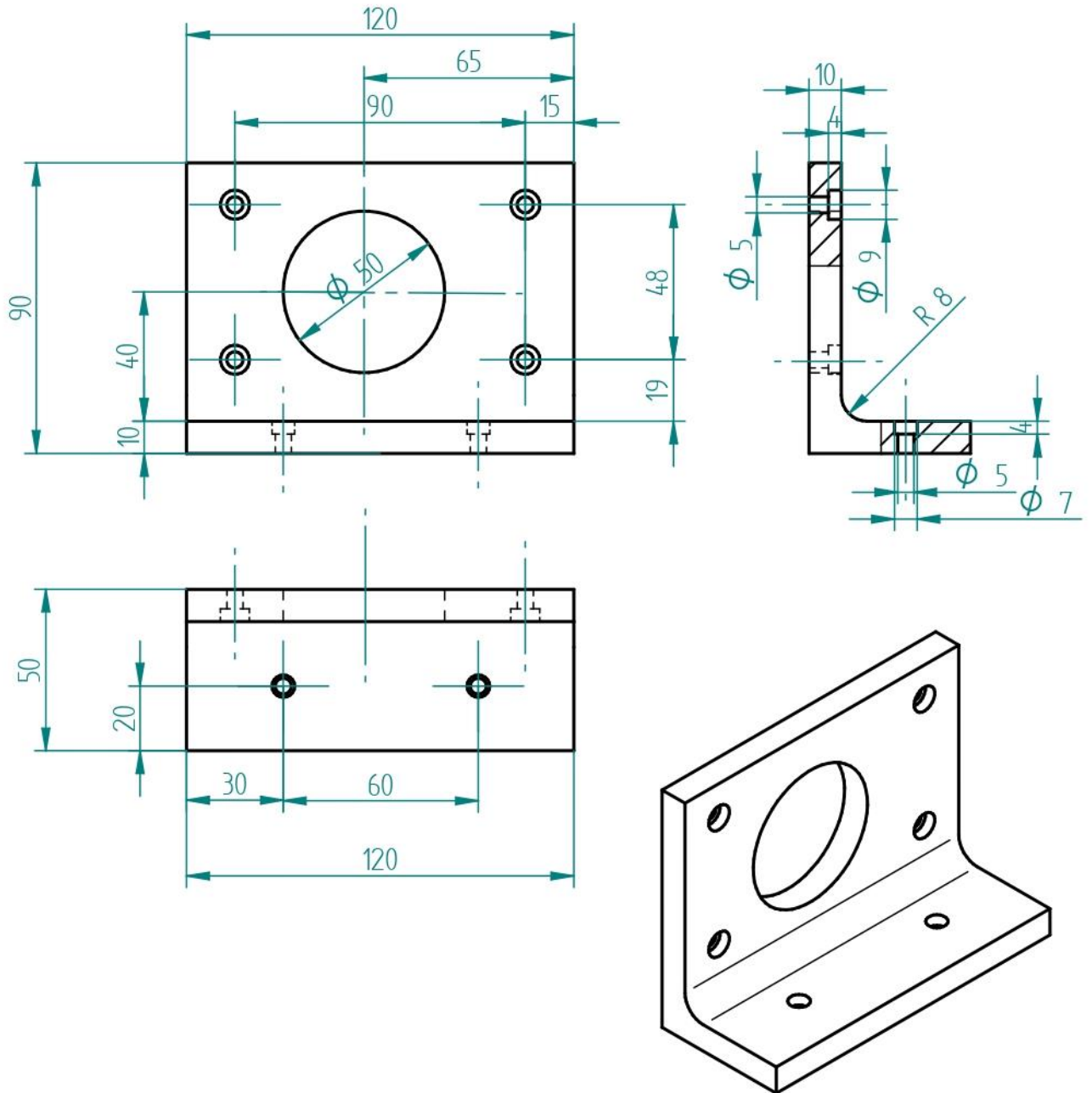
<https://github.com/uriguen95/pilotaaltuerarenkontrola.git>


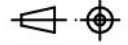
## ERANSKINAK 3: Planoak

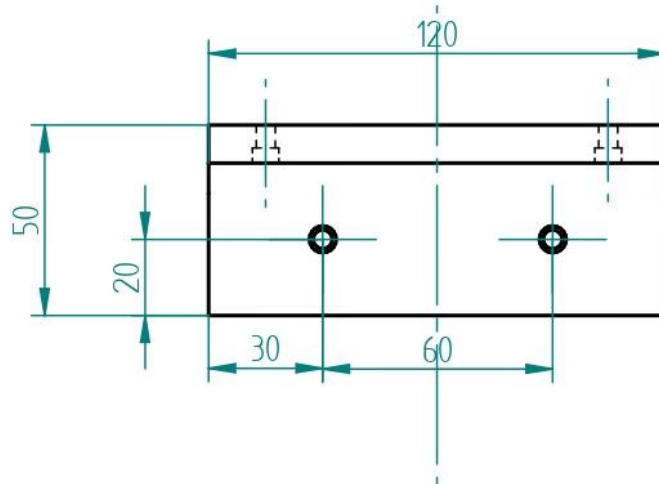
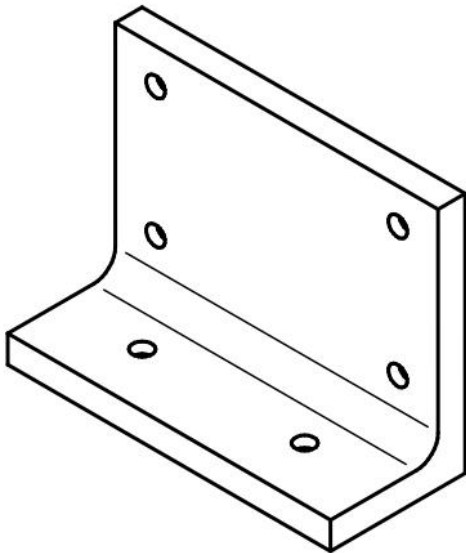
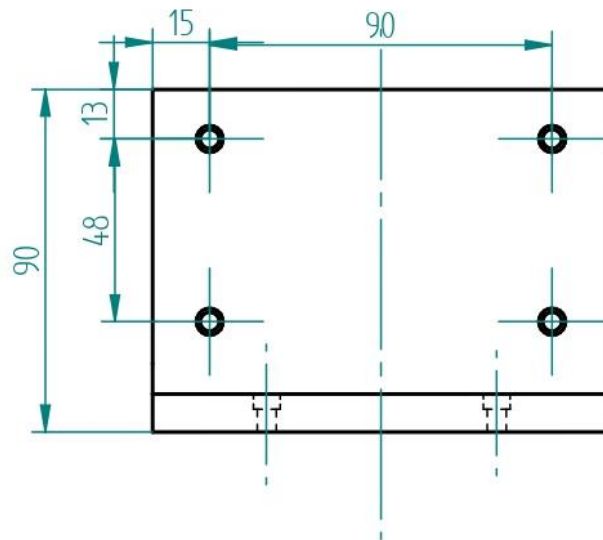
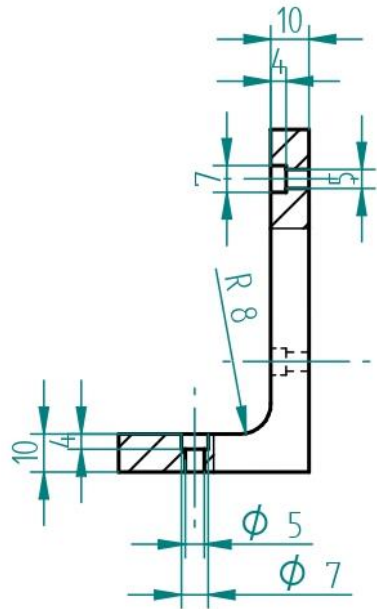
3D inpresioaren bitartez fabrikaturiko piezen planoen sailkapena:


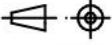
Taula 0-1 Inprimaturiko piezen planoen sailkapena

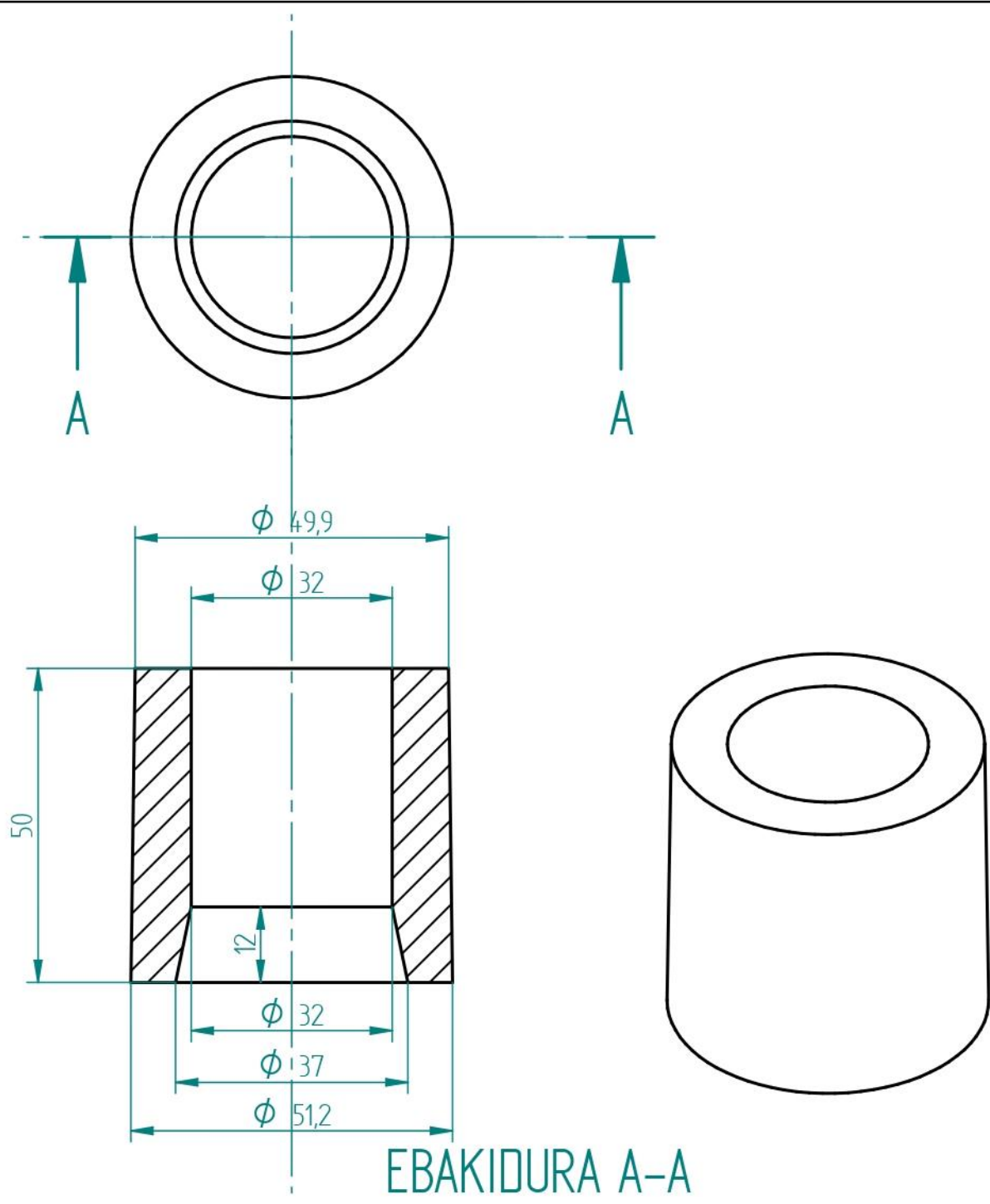
Plano Zenbakia	Deskribapena	Formatua	Eskala
P1	Zulodun Oinarria	A4	2:1
P2	Zulorik Gabeko Oinarria	A4	2:1
P3	Konexioa	A4	2:1



1	Zulodun Oinarria	1		PLA		
Pieza Kop.	Izendapena eta Oharrak	Marka	Araudia Planoa	Materiala	Bak.	Osoa
					Pisua	
		Data	Izena	 <b>EUSKAL HERRIKO UNIBERTSITATEA</b> BILBOKO INGENIARITZA ESKOLA		
Marraztua:	19/03/15	Ekain Uriguen				
Gainbegiratua:	19/07/16	Oskar Casquero				
 Perdoi Orok.	Eskala	<b>Zulodun Oinarria</b>		<b>PILOTA BATEN ALTUERAREN KONTROLA ARDUINO ETA MATLAB BITARTEZ</b>		
12	Plano Zkia. : 1					
	Plano Kop. : 3					



1	Zulorik Gabeko Oinarria	1		PLA		
Pieza Kop.	Izendapena eta Oharrak	Marka	Araudia Planoa	Materiala	Bak. Pisua	Osoa
Marraztua:	19/03/15	Ekain Uriguen	 <b>EUSKAL HERRIKO UNIBERTSITATEA</b> BILBOKO INGENIARITZA ESKOLA			
Gainbegiratua:	19/07/16	Oskar Casquero				
 Perdoi Orok.	Eskala 1:2	<b>Zulorik Gabeko Oinarria</b>		<b>PILOTA BATEN ALTUERAREN KONTROLA ARDUINO ETA MATLAB BITARTEZ</b>		
				Plano Zkia. : 2		
				Plano Kop. : 3		



1	Konexioa	1		PLA		
Pieza Kop.	Izendapena eta Oharrak	Marka	Araudia Planoa	Materiala	Bak.	Osoa
					Pisua	
	Data	Izena	 EUSKAL HERRIKO UNIBERTSITATEA BILBOKO INGENIARITZA ESKOLA			
Marraztua:	19/03/15	Ekain Uriguen				
Gainbegiratua:	19/07/16	Oskar Casquero				
 Perdoi Orok.	Eskala	Konexioa		<b>PILOTA BATEN ALTUERAREN KONTROLA ARDUINO ETA MATLAB BITARTEZ</b>		
	1:2			Plano Zkia. : 3		
				Plano Kop. : 3		