

GRADUA: Industria Teknologiaren Ingeniaritzako
Gradua

GRADU AMAIERAKO LANA

TENPERATURA MAKETA BATEN IDENTIFIKAZIOA ETA KONTROLA

Ikaslea: Lopez de Zubiria Erostarbe, Jon

Zuzendaria : Iriondo Urbistazu, Nagore

Ikasturtea: 2017-2018

Data: Bilbon, 2018ko uztailaren 17a

eman ta zabal zazu



Universidad del País Vasco Euskal Herriko Unibertsitatea

BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO

[EU] Laburpena

Proiektu honen helburua tenperatura maketa erreal baten kontrol-sistema diseinatzea da. Horretarako, lehendabizi sistemaren eredua lortzeko software bat eraiki da, ondoren eredu horretatik kontrolagailuaren diseinua egin ahal izateko eta azkenik aurretik kalkulaturako diseinu parametroekin dagokion kontrol-sistema implementatu ahal izateko programa bat sortu da. Maketaren identifikazioa eta ondorengo kontrola PLC baten bidez egin da.

[EN] Abstract

The aim of this project is to design the control system of a heat flow experiment. To achieve this aim, first we must create a software for the identification of the system to later on make the design of the controller and finally with the parameters calculated for the controller we need a software implement the control. For the identification and the control of the heat flow experiment the tool that has been use is a PLC.

[ES] Resumen

El objetivo de este proyecto es diseñar un sistema de control que actúe sobre una maqueta de temperatura real. Para ello, primero se ha creado un software para conseguir un modelo aproximado del sistema, para así después con el modelo poder diseñar el controlador y finalmente implementar un sistema de control sobre la maqueta. Para la identificación y control de la maqueta se ha dispuesto de un PLC.

AURKIBIDEA

1. SARRERA ETA TESTUINGURUA	5
2. HELBURUA ETA IRISMENA	8
3. LANAREN ONURAK	10
4. AUKEREN ANALISIA	11
4.1. AUKERA METODOLOGIKOAK	11
4.1.1. EREDUAREN IDENTIFIKAZIOA	11
4.1.2. KONTROLAREN DISEINUA	12
4.1.3. PID KONTROLAGAILUAK	13
4.2. HARDWARE ETA SOFTWARE ALTERNATIBAK	15
5. METODOLOGIA	16
5.1. MAKETAREN DESKRIBAPENA	16
5.2. SEINALEEN TRATAMENDUA	19
5.3. TIA PORTAL-EN SOFTWARE ARKITEKTURA	20
5.3. PROZESUAREN EREDUA	21
5.3.1. SOFTWARE DISEINUA ETA INPLEMENTAZIOA	22
5.4. KONTROL ZUZENA	28
5.4.1. SOFTWARE DISEINUA ETA INPLEMENTAZIOA	28
6. APLIKAZIOAREN ADIBIDEA	32
6.1. PROZESUAREN EREDUA LORTZEKO ESPERIMENTUA	32
6.2. KONTROLAGAILUAREN DISEINUA	33
6.3. KONTROLAGAILUAREN BALIZKOTZEA	34
6.4. ESPERIMENTUAREN ANALISIA	37
7. BETEBEHARREN DESKRIBAPENA	38
7.1. GANTT DIAGRAMA	39
8. AURREKONTUA	40
9. ONDORIOAK	42
INFORMAZIO ITURRIAK	43
ERANSKINAK I: IDENTIFIKAZIORAKO PROGRAMAREN KODIGOA	44
ERANSKINAK II: BEGIZTA ITXIKO KONTROLAREN KODIGOA	45

IRUDIEN AURKIBIDEA

Irudia 1: SIMATIC S7-1500 eta Quanser Heat Flow Experiment	6
Irudia 2: PID ekuazio klasikoa	14
Irudia 3: PID ez interatiboaren ekuazioa.....	14
Irudia 4: PID ez-interatiboaren diskretizazioa.....	14
Irudia 5: Maketaren irudia sarrera eta irteerak adierazita	16
Irudia 6: Fabrikatzaileak emandako sentsoreen ezaugarriak. Temperatura °C-etan eta tentsioa V-etan.....	17
Irudia 7: Berogailuaren eta sentsorearen tentsioaren arteko erlazioa.	18
Irudia 8: PLC-aren sarrera eta irteera analogikoen konfigurazioa.	19
Irudia 9: SCALE eta UNSCALE funtzioak ulertzeko irudia.	20
Irudia 10: Prozesuaren eredua lortzeko eskema.	21
Irudia 11: Software arkitekturaren eskema.....	22
Irudia 12: FB-aren funtzionamenduaren irudi eskematikoa.....	23
Irudia 13: Eragingailuaren sarrerako pulsu karratua.....	23
Irudia 14: Eredua lortzeko programaren FB-a.....	25
Irudia 15: Prozesuaren irudiaren OB30-aren konfigurazioa.....	25
Irudia 16: Prozesuaren irudiaren Traces konfigurazioa.	26
Irudia 17: Prozesuaren irudiaren Traces konfigurazioa.	27
Irudia 18: Kontrol-sistema baten eskema.....	28
Irudia 19: Kontrol zuzenaren FB-aren irudi eskematikoa.....	29
Irudia 20: Kontrol zuzenaren FB-aren sarrerak eta irteerak.	30
Irudia 21: Kontrol zuzeneko OB30-aren konfigurazioa.	31
Irudia 22: Kontrol zuzeneko Traces-aren konfigurazioa.	31
Irudia 23: OB30-etik egindako instantzia,	32
Irudia 24: Traces-eki lortutako 1.go sentsorearen grafikoa. Y ardatzean temperatura °C-tan eta X ardatzean denbora segundotan.	33
Irudia 25: Kontrol zuzeneko OB30-ari eginiko instantzia.....	35
Irudia 26: Traces-en neurketa, Sentsorea eta Setpoint-a, temperatura °C-etan denboraren funtzioan segundotan.....	36
Irudia 27: Traces-en neurketa, berogailuaren kitzikadura denborarekiko, tentsioa V-etan denbora segundotan.	36
Irudia 28: Betebeharren Gantt diagrama	39

AKRONIMOEN ZERRENDA

DB Data Block

FB Function Block

FBD Function Block Diagram

FC Function

IL Instruction List

LD Ladder Diagram

OB Organization Block

PC Personal Computer

POU Program Organization Unit

ST Structured Text

TIA Portal Totally Integrated Automation Portal

EHU Euskal Herriko Unibertsitatea

1. SARRERA ETA TESTUINGURUA

Ingalaterran, XVII. Mendean, Industria Iraultzak sekulako aurrerapen teknologiko eta ekonomikoa suposatu zuen gizarterako. Aurretik zeuden esku-lanabesak, animalien trakzio-indarrak, industria eta merkantzia zein pertsonen garraiorako makinengatik ordezkatuak izan ziren. 1774. urtean James Watt-ek asmatu zuen lurrun-makina XVIII mendeko asmakizun garrantzitsuenetakoa izan zen, Industria Iraultzaren ardatz nagusia izan baitzen.

Nahiz eta makina industrialen indarraz baliatu, ezin bestekoa izan zen langileen esku-hartzea prozesu guzti horiek aurrera eraman ahal izateko, oraindik industriaren automatizazioa oso urrun baitzegoen. Testuinguru horretan, langileen espezializazioa eta muntaketa-kateen eratzea sortu zen. Henry Ford izan zen muntaketa kateen inplementazioa aurrera eraman zuen lehena, prozesu errepikakorrenzat oso aproposa zena eta etorkizuneko automatizaziora aurrerapausoa izan zen.

Automatizazioa industrialara ekoizte, egokitze, banatze, aztertze eta kudeatze prozesuetan gizakien esku-hartze zuzena murriztea ahalbidetzen duten tekniken multzoa da. Automatizazio tekniken euskarri teorikoak automatika (matematika aplikatuen adarra) izeneko zientziatik datoz. Prozesu baten eragiketa gizakiaren esku-hartzea gabe gauzatzen denean honi erregulazio automatikoa deritzo. Hau automatizazio industrialaren muina dela esan daiteke, eta hauek dira oinarritzko printzipioak: egonkortu beharreko magnitude bakoitza neurtu, neurketaren bat-bateko balioa igorpen balioarekin alderatu, bereizketa-seinale bat eratu, seinalea funtzio sorgailu batean tratatu eta aginte seinalea sortu, horri dagokion erregulatze-organoak gida dezan.

Kontrolaren inplementaziorako gaur egun industrian oso hedatuak dauden PLC-ak. Dispositibo hauek, prozesu logiko errepikakorrak kontrolatzen dituzten automatikak dira. Merkatuan PLC-en eskaintza handia dago: Siemens, Hitachi, GE-Fanuc... Mende honetararte fabrikatzaile bakoitzak bere programazio-lengoaia garatu du, ondokoari erreparatu gabe. Honek erabiltzailearentzat traba bat suposatu du, batetik, lengoaia desberdinak ikasi behar zituelako eta bestetik marka desberdineko gailuen artean bateragarritasun arazoak zeudelako.

Testuinguru honetan, 2000. urtetik aurrera software-eraikuntzarako estandarrak finkatzen hasi ziren emaitza IEC 61131 estandarizazioa izanik. Zehatzagoak izateko estandar honen hirugarren atalean, IEC 61131-3, programazio-lengoaiei

estandarizazioari buruz hitz egiten da lengoia-mota desberdinen estandarizazioa finkatuz: IL, LD, ST eta FBD .

Kontrola inplementatzeko ingurunea “Bilboko Ingeniaritza Eskola”-ko “Sistemen Ingeniaritza eta Automatika” departamentuan eskuragai dagoen Siemens-en SIMATIC S7-1500 PLC-a izan da. Software aldetik, ingurune honekin bateragarria den TIA Portal V13 erabili da. Siemens automaten munduan izen onekoa eta estandarizazio altuko softwarea eskaintzen duena. Programazio lengoian sartuz, proiektu honetan ST motako Siemens-ek sortutako SCL-a erabili da. Lengoai hau goi mailakoa da eta programatzerako garaian erraza.

Automatizazio eta kontrolaren prozesuetara itzuliz, bi prozesu-mota bereiz daitezke: prozesu diskretuak eta prozesu jarraituak. Prozesu diskretuak gehien bat logika boolearraren bidez kontrolatzen direnak dira, adibidez: muntaketa kate automatiko bat. Bestalde, prozesu jarraituak ditugu hauetan sistemaren aldagai baten kontrola egiten da. Aldagaiaren kontrola egiteko periodo bakoitzeko aldagaiaren neurketa eta erreferentziatzeko balioaren arteko konparaketa egiten da eta honen arabera kontrolagailu eragingailuak eman behar duen erantzuna kalkulatu du.

Kasu honetan, sistema jarraitu baten kontrola diseinatu da, temperatura maketa batena. Proiektuan Quanser markako “Heat Flow Experiment” maketarekin lan egin da. Temperatura maketa kamara termiko bat da, non haizegailu eta berogailu batekin temperatura erregulatu daitekeen. Bestalde, toki desberdinetan kokatuta dauden hiru sentzore ditu temperatura neurtzeko. Maketak prozesu erreala eskala txikiago batean gauzatzeko tresnak dira eta hauen kontrol-sistema diseinatzeko kurtsoan zehar “Automatika Industrial” eta “Konputagailu bidezko Kontrola” ikasgaietan bereganatutako ezagutzak aplikatu dira.



Irudia 1: SIMATIC S7-1500 eta Quanser Heat Flow Experiment

Maketaren kontrol zuzena egiteko PID kontrolagailu bat erabili da. PID kontrolagailuak hiru akzio dituzten kontrolagailuak dira (proporzionala, integrala eta diferentziala) eta errorearen araberako seinalea ematen. Gaur egun prozesu industrialetan gehien erabiltzen diren kontrolagailuak dira.

2. HELBURUA ETA IRISMENA

Proiektu honen helburu nagusia temperatura maketa baten kontrol-sistema diseinatzea eta inplementatzea da, horretarako Siemens-en S7-1500 PLC-az baliatuz. Modu honetan, maketa erreal batekin lan egin da errealitatean agertzen diren alderdi praktikoak lantzeko.

Kontrol-sistema bat diseinatzerakoan helburu partzial bezala finkatu diren faseak bete behar dira:

1. Portaeraren dinamika ezagutzea: Honetarako eredu matematikoak erabiltzen dira. Eredu matematikoa sistemaren aldagai nagusien eta operazio puntuaren arabera da, ez dago eredu matematiko bakarrik maketaren portaera bere osotasunean adieraziko duena. Temperatura-maketaren kasuan bi eragingailu daude (berogailua eta haizegailua) eta hiru sentore. Sistema bat definitzeko aldagai manipulatu bat, eragingailua, eta aldagai kontrolatu bat, sentorea, aukeratu behar dira. Honekin batera, operazio puntua aukeratu behar da, hau da, zer temperatura tartetan egongo den sistema. Geroz eta operazio puntutik gertuago egon orduan eta baliagarriagoa izango da sistema.
2. Kontrolagailuaren diseinua: Fase honetan ereduaren azterketaren bidez kontrolagailuaren aukeraketa eta kontrol parametroen sintonizazioa egiten da. Parametroak kalkulatzeko metodo eta kriterio asko daude. Helburua sistemak begizta itxiko kontrolean ezartzen diren espezifikazioak betetzea da.
3. Kontrol zuzena: azken fasea kontrolaren inplementazioa da. Aurretik kalkulatuak parametroak kontrolagailuan, kasu honetan PID-an, sartu eta kontrola exekutatu da. PID kontrol-sistema erroreak aurrean erantzuten duen sistema bat da, beraz, etengabe aldagai kontrolatuaren eta erreferentziaren arteko konparaketa egin behar da erroreak arabera kontrol akzioa kalkulatzeko. Honi begizta ixtea deitzen zaio, kontrol-sistema guztien oinarria dena. Temperatura-maketaren kasuan, bi eragingailu dituen sistema berogailua edo haizegailuaren bidez kontrolatu daiteke, eta, era berean, hiruetako edozein sensorerekin neurtu daiteke temperatura. Sentoreen kokapenagatik, neurketen artean atzerapena eman daiteke.

Kontrol-sistemaren inplementazioan zailtasun nagusia datuen tratamenduan dago, seinale elektrikoak kudeatzeko garaian. PLC baten bidez egingo denez, berogailu edo haizegailuan sartu nahi dugun potentziari baliokidea zaion tentsioa atera behar da PLC-aren irteera analogikotik, eta era berean, sentsoreak igortzen duen tentsioa tenperatura batekin identifikatu behar da.

TIA Portal V13 softwarearen bidez egin da programa, ingurune honek estandarizazio handiko kodigoa erabiltzen du eta horregatik parametro formalak dituzten bloke funtzionalak erabili dira. Era honetan, diseinatu dugun kontrol-sistema tenperatura-maketaren edozein sisteman zein beste edozein prozesu jarraitu batean aplikagarria izan daiteke.

3. LANAREN ONURAK

Lehen esan bezala, maketak prozesu errealak eskala txikiago batean egiteko erabiltzen dira. Honela eskala handiago bateko prozesu baten azterketa egin daiteke, espazio murriztuago batean. Honek operazio kostuak murrizten ditu, azken finean laborategiek espazio murriztua dute eta eskala errealeko prozesu bat gauzatzeak kostu ekonomiko handiagoa dauka.

Beste onuretako bat proiektu honetan erabili den sistemaren identifikaziorako metodoa da. Lehen esan bezala, identifikaziorako metodo experimentalaren bidez kontrolatutako prozesuaren, kasu honetan tenperatura maketaren, portaera erreala adieraziko digun eredu matematikoa lortuko dugu. Gainera, proiektu honetan garatutako identifikaziorako metodo hau ez da soilik tenperatura-maketara mugatzen, beste maketa edo sistemetan erraz aplikagarria da.

Lehen errepikatu dugun bezala, maketaren edo beste edozein sistemaren ereduia izatea ezinbestekoa da edozein kontrol diseinatzerako garaian. Eredua edukitzeak prozesuaren simulazioa egitea ahalbidetuko dugu, eragiketak izango duen erantzuna ezagutzuz prozesua martxan jarri baino lehen.

Beste alde batetik, TIA Portal programaren inguruan ezagutzak bereganatzen lagunduko digu. Ez da ahaztu behar makina hauen konplexutasuna, industrian erabiltzen diren makina profesionalak baitira, eta praktikan arazo ugari ematen dituzte bereziki erabiltzaileak esperientzia eskasa badu.

Programa eraikitzerako garaian, gehien kontuan hartu den ezaugarria berrerabilpena izan da. Honekin esan nahi dena da, diseinatu den kontrol sistema honekin beste prozesu batzuk ere kontrolatzeko balio duela. Horretarako, parametro formalak dituzten bloke funtzionaletan antolatu da programa, non hauek nahi adina instantziatu daitezkeen beste sistema-motetan kontrola inplementatuz ahalik eta zailtasun gutxienarekin.

Beraz hemen erabilitako programak eta kodigoak etorkizuneko beste proiektu batean berrerabiltzeko aukera ematen du.

4. AUKEREN ANALISIA

Atal honetan temperatura maketaren kontrola gauzatzeko aukera desberdinen analisia egin da. Alde batetik, alternatiba metodologikoak ditugu bertan maketaren ereduaren identifikaziorako, kontroladorearen diseinurako eta PID kontrolerako aukera desberdinen inguruan hausnartuko da. Bestetik, erabilitako erreminten analisia egingo da bi alorretan: software eta hardware.

4.1. AUKERA METODOLOGIKOAK

Lehenago azaldu bezala atal honetan ereduaren identifikaziorako, kontrolagailuaren sintonizaziorako eta PDI kontrolerako desberdinak azalduko dira.

4.1.1. EREDUAREN IDENTIFIKAZIOA

Ereduaren identifikaziorako bi metodo desberdinak dira, bata metodo teorikoa eta bestea metodo esperimentalak.

- **Metodo teorikoa:** Sistemaren identifikazioa modu teorikoan egiten da. Horretarako, sistemaren dinamika ezagutu behar da eta lege fisiko eta energia kontserbazio legeak aplikatuz tenperaturaren ekuazio diferentzialak lortzen dira. Metodo honen aplikaziorako maketaren aspektu asko ezagutu eta kalkulatu behar dira: material, beroarekiko iragazkortasuna, ingurugiroko tenperatura, berogailuaren efizientzia... Gainera energiaren kontserbazio legea aplikatzea ez da triviale eta eredu konplexuak lortzen dira. Beraz metodo hau baztertuko da.
- **Metodo esperimentalak:** Sistemaren eredu esperimentu baten bidez lortzen da. Horretarako, manipulatuak aldagaiekin aldaketa bat sortzen da aldagai manipulatuak eragiteko. Aldagai kontrolatuaren erantzuna erregistratzen da handik eredu matematikoa lortzeko. Metodo honekin sarrera eta irteeraren azterketa eginez eredu sinpleak identifikatzen dira.

4.1.2. KONTROLAREN DISEINUA

Sistemaren eredua dugularik, kontrolagailuaren aukeraketara eta ondorengo parametroen sintonizaziora pasa gaitzke. Sintonizazioari buruz hitz egiten denean, sistemaren begizta irekiko eredua aztertzeari eta kontrol begizta ixterakoan espezifikazio zehatz batzuk betetzeko kontrol parametroen kalkuluari buruz dihardugu. Parametro hauen kalkulurako metodo desberdinak daude, hauen aukeraketa sistema-motaren arabera, erabiltzailearen aukerakoak eta kontrol motaren arabera dira. Halere, atal honetan hiru metodo nagusi aztertu dira (esperimentala, autosintonizazioa eta analitikoa) eta aukeraketari buruzko gomendioak eman dira.

- **Esperimentala:** modu enpiriko batean lortutako taulak aplikatzean datza. Horretarako sistemaren dinamikaren arabera eta kontroleko espezifikazioen arabera tauletako parametroen aukeraketa egiten da. Sintonizazio esperimentala asko erabiltzen da industrian eta metodo asko daude: Ziegler-Nichols, Amström-Hägglund, metodo integrala... Metodo esperimentala erabiltzeko atzerapena duen lehen ordenako sistema bat behar da.
- **Autosintonia:** Kontrol-sistema batzuk parametro hauek modu automatikoki kalkulatu dituzte optimizazio fasea batean.
- **Analitikoa:** metodo hau sinpleena da, batez ere, lehen ordenako sistementzat. Sistema irekiko polo eta zeroen analisia egin ondoren kontrolagailuari parametro batzuk ematen zaizkio begizta itxiko sisteman eragiteko.

Esperimentuetan lortu den esperientziaren ondorioz, tenperaturak lehen ordenako sistema baten portaera duela ondorioztatu da, beraz, aurrerago azalduko den bezala begizta irekiko esperimentuaren adibidea egiterakoan metodo analitikoa aukeratu da.

4.1.3. PID KONTROLAGAILUAK

Sistemaren kontrola egiteko TIA Portalek baditu aurretik eginak dauden kontrolagailuak: PID_Compact, PID_3Step eta PID_Temp. Bestetik, norberak diseinatutako PID algoritmoa inplementatu daiteke, kasu honetan PID ez interatiboa diskretizatu eta inplementatu da.

- **PID_Compact:** PID erreguladore jarraitua da optimizazio integratuarekin. Magnitude ugari erregulatzeko balio du adibidez: Tenperatura. Gehienetan, erregulazio noranzko normalean erabiltzen da, hau da, irteeraren gorakadak magnitude kontrolatuaren gorakada dakar. Adibidez: kalefakzio baten potentzia igotzean tenperaturak ere gora egingo du.

PID_Compact-ek P, I eta D parametroen aukeraketa modu automatikoan egiten du hasierako optimizazio fasean. Parametro hauek gehiago optimizatu daitezke 'optimización fina' aukera sakatuz, ez da beharrezkoa parametro hauek modu manualetan aukeratzea.

- **PID_3Step:** Objektu teknologiko balbula eta eragingailu integratzaileentzat PID erreguladore bat da. Aurreko PID-a bezala gehienetan erregulazio noranzko normalean erabili ohi da. Kontrolagailu honek ere P, I eta D parametroen aukeraketa automatikoa egiten du.
- **PID_Temp:** Objektu teknologiko honek PID erreguladore jarraitua optimizazio integratuarekin eskaintzen du. PID_Temp-a bereziki tenperaturaren erregulaziorako diseinatua dago eta bai kalefakzio bai kalefakzio/errefrigerazio sistemetan aplikatu daiteke.

Kontrolagailuak P, I eta D parametroak modu automatikoan aukeratzen ditu hasierako optimizazio prozesuan. Optimizazio zehatzagoa nahi bada "Optimización fina" sakatu behar da.

Kalefaktore/errefrigeradore aplikaziorako errefrigerazio faktore bat aukeratu daiteke edo bi PID joku erabili.

- **PID ez interatiboa:** Kontrol algoritmo ez-interatibo bezala izendatzen dira 2. Irudiko PID ekuazio klasikotik datozenak. Ez-interatibo kalifikatzaileak adierazten duena akzio integrala eta diferentziala elkarrekiko independenteak direla, naiz eta irabazpen proportzionalak K_c hiru akzioetan (proportzionala, integrala eta diferentziala) eragiten duen.

Algoritmo hau estandar bezala kontsideratzen da ISA-ren (Instrument Society of America) eta gaur egunean erabiliena da.

$$u(t) = K_c \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) = P + I + D$$

Irudia 2: PID ekuazio klasikoa

PID-a eraikitzeke erabili den ekuazioa 3. Irudiak erakusten duen PID errealaena da. PID ez-interatibo idealak maiztasun handietan akzio diferentzialak irabazpen altuegia hartzen du, beraz hori ekiditeko denbora konstantea T_d/N duen lehen ordenako sistema batekin filtratzen da. N -ren balio ohikoenak 2-20 artekoak dira.

$$U(s) = K_c \left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d s}{N}} \right) E(s)$$

Irudia 3: PID ez interatiboaren ekuazioa

PID-a inplementatzeko ekuazio diferentziala diskretizatu behar da eta algoritmoa exekutatu duen FB batean sartu.

$$u_p(k) = K_c \cdot e_p(k)$$

$$u_i(k) = u_i(k-1) + K_c \cdot \frac{T}{2 \cdot T_i} \cdot [e_i(k) + e_i(k-1)]$$

$$u_d(k) = \frac{u_d(k-1) \cdot T_d + K_c \cdot T_d \cdot N \cdot [e_d(k) - e_d(k-1)]}{(N \cdot T + T_d)}$$

$$u(k) = u_p(k) + u_i(k) + u_d(k)$$

Irudia 4: PID ez-interatiboaren diskretizazioa

4.2. HARDWARE ETA SOFTWARE ALTERNATIBAK

Maketaren gain eragiteko, bai begizta irekiko esperimientua egiteko zein PID bidezko kontrola egiteko, dispositibo programagarri bat behar da. Testuinguru honetan, nahiz eta aurreko ataletan horretarako erabili dugun erreminta PLC bat izan, beste alternatiba bat ere planteatu egin da: PC baten bidezko kontrola. Hardware-alternatiba bakoitzak bere softwarea erabiltzea dakarrenez, analisi hau modu independente batean ezin dela egin ulertu da.

- **PC:** Gaur egun ordenagailu pertsonalek malgutasun handia eskaintzen dute eta kasu honetan bezalako sistemetan kontrol bat ezartzea guztiz posiblea da.

Tresna hau erabiltzerako garaian maketa errearen gain eragin nahi bada sarrera eta irteera analogikoak irakurriko dituen tarjeta baten beharra dago. Software aldetik, Matlab/Simulink-en toolbox-a den RTW(Real Time Window) tarjet erabili behar da sarrera eta irteera analogikoen tarjetarekin konektatzeko.

- **PLC:** Aurretik aipatu bezala PLC-ak ahalmen handiko gailuak dira, prozesu bat baino gehiago gauzatu dezakete aldi berean eta gaur egungo industrian oso hedatuak. PLC-en abantaila bat txartel analogikoa integratu dezaketela da. PLC-en artean marka asko daude, Beckhoff eta Siemens besteak beste. Siemens-en SIMATIC S7-1500 aukeraketaren arrazoiak alde batetik marka fidagarria eta eskarmentu handikoa dela eta bestetik, eskolan eskuragarri dugun tresna dela.

Software aldetik, TIA Portal V13-a erabili da. Software honen abantaila hardware-arekin bateragarria izateaz gain estandarizazio altukoa dela da, IEC-61131-3aren arabera.

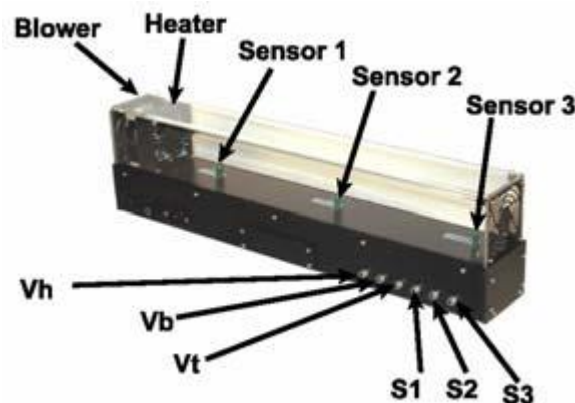
5. METODOLOGIA

Atal honetan proiektua gauzatzeko metodologia azalduko da. Horretarako lehendabizi prozeduren ulermenean lagunduko duten erabilitako hardware eta software plataformei buruzko aspektu batzuk azalduko dira eta ondoren maketaren eredua lortzeko eta PID kontrola inplementatzeko eraiki diren programak azalduko dira.

5.1. MAKETAREN DESKRIBAPENA

Software-arkitekturarekin hasi baino lehen ezinbestekoa da temperatura-maketaren ezaugarriak ezagutzea: sarrerak, irteerak, kalibrazioa...

Maketak bi sarrera ditu, bata berogailua eta bestea haizegailua. Bien potentzia maximoa berdina da, 400 w. Tentsioari erreparatuz, 0-5v arteko tentsioaren kitzikatu daitezke bakarrik.



Irudia 5: Maketaren irudia sarrera eta irteerak adierazita

Irteerei erreparatuz, hiru sentsore ditugu (S1,S2 eta S3), 5.irudian ikusten den bezala, berogailutik gertuenetik urrunenera ordenatuta. Sentsoreek sumatzen duten temperatura seinale elektriko batera pasatzen dute. Seinale hau 0-5 V artekoa da eta sentsoreen irabazpena 20°C/V, hau da, 5 V-etako seinale bat emititzen dutenean 100°C-tara daudela esan nahi du.

<i>T</i>	19	20	21	22	23	24	25	26	27	28	29	30
<i>V</i>	0,95	1,00	1,05	1,10	1,15	1,20	1,25	1,30	1,35	1,40	1,45	1,50

Irudia 6: Fabrikatzaileak emandako sentsoreen ezaugarriak. Temperatura °C-etan eta tentsioa V-etan.

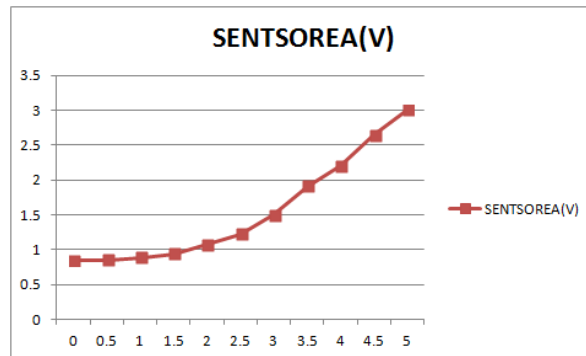
Kasu honetan, tenperaturen eta berogailuan sartzen dugun kitzikadura-tentsioaren arteko harremana aztertuko dugu. Esperimentu hau oso garrantzitsua da sistemaren linealtasun-eremua aztertzeko, eremu honetan ibili beharko dugu sistemaren erantzun egoki bat eskuratzeko.

Azterketa hau modu experimental batean egin da. Berogailuaren sarrera tentsioa 0-5 V tartean egonik, bitarte horretan 0.5 V-etako tartearekin kitzikatu da aldiro. Aldi bakoitzean, sistema egonkortzen utzi eta sentsoreak emandako tenperatura neurtu da.

Esperimentuan lortutako datuak excel orri batean jarri eta gero emaitza hauek lortu dira:

BEROGAILUAREN TENTSIOA(V)	SENTSOREA(V)	SENTSOREA(T)
0	0,84	16,8
0,5	0,85	17
1	0,89	17,8
1,5	0,94	18,8
2	1,07	21,4
2,5	1,22	24,4
3	1,5	30
3,5	1,91	38,2
4	2,2	44
4,5	2,64	52,8
5	3	60

Taula 1: Berogailuaren tentsioa eta sentsorearen tenperatura eta tentsioaren arteko erlazioa



Irudia 7: Berogailuaren eta sentsoarearen tentsioaren arteko erlazioa.

Datu hauetatik zenbait ondorio atera ditugu. Alde batetik, prozesua lineala izango da berogailuaren sarrera tentsioaren 2-5 V-etako balio-tarterako. Horrek esan nahi du, 0-2 Voltetako tentsioarekin kitzikatzen bada 7. irudian ikusten den bezala tenperaturaren aldaketa oso txikia izango da.

Bestetik, tenperaturaren tarte ikus daiteke. Berogailua martxan ez dagoenean maketaren tenperatura inguru girokoa izango da (16,8°C) eta berogailuaren potentzia maximoarekin tenperaturara 60°C-tara irits daiteke. Esperimentu honen emaitza ingurugiroko baldintzei atxikituta dago, hau da, esperimentua gauzatzen den gelaren tenperaturaren arabera izango da.

Honekin, esperimentuak egitean, maketaren mugak kontuan hartu ahal izango dira.

5.2. SEINALEEN TRATAMENDUA

Programaren eraikuntzan erronketako bat datuen tratamendua da. Seinale elektrikoekin lan egiten denez hauen karakterizazioa ezinbestekoa da datuen kudeaketa egokia izateko.

Alde batetik, PLC-ak igortzen eta jasotzen dituen seinale elektrikoak daude. PLC-aren sarrera eta irteerako moduluak $-10/+10$ V arteko seinaleak jasotzeko eta igortzeko konfiguratu dira. PLC-aren sarreran seinale elektriko bat sartzen denean, moduluaren txartelak $-27648/+27648$ arteko eta proportzionala den seinale analogiko batera pasatzen du. Adibidez, sarrera analogikotik 5 V sartzen badira, sarrera moduluak 13824 seinale bitar bezala jasoko du. Era berean, PLC-tik 5 V-tako seinale bat atera nahi badugu, 13824 seinale bitarra zuzendu behar da. 8. Irudian PLC-aren sarrera eta irteera analogikoen konfigurazioa ikus daiteke hardwarea konfiguratzearterako analogikoen parametrizazioari dagokiona.

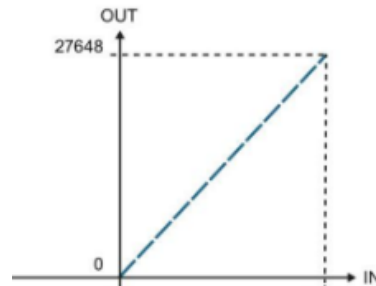
The image shows a software window titled "Medición" (Measurement) with several configuration options for an analog input:

- Tipo de medición: Tensión
- Rango de medición: +/- 10V
- Coefficiente de temperatura: +/- 50mV, +/- 80mV, +/- 250mV, +/- 500mV
- Unidad de temperatura: +/- 1V, +/- 2.5V, +/- 5V
- Unión fría: +/- 10V (highlighted), 1..5V
- Temperatura de referencia fija: 1..5V
- Supresión de frecuencias perturbadoras: 50 Hz
- Filtrado: Ninguno

Irudia 8: PLC-aren sarrera eta irteera analogikoen konfigurazioa.

Sarreren eta irteeren karakterizaziorako eragingailuen eta sentsoreen ezaugarriak ezagutu behar dira. Bi eragingailu eta hiru sentsore daude eta haien ezaugarriak 6.1. atalean azaltzen direnak dira. Seinale baten karakterizaziorako TIA Portal-ek duen UNSCALE eta SCALE funtzioak erabili behar dira.

Sarrera bat dugunean seinale hori identifikatzea izango da, hau da seinale analogiko horri dagokin balio erreala kalkulatzeko. Kasu honetan seinale analogikoari dagokion tenperatura kalkulatu behar da. Horretarako, SCALE erreminta erabiltzen da. SCALE funtzioak hiru sarrera ditu: eskalatu behar den seinale analogikoa, seinale errearen balio maximoa, 27648 balio bitarrari baliokidea dena, eta seinale analogikoaren balio minimoa.



Irudia 9: SCALE eta UNSCALE funtzioak ulertzeko irudia.

SCALE-ren funtzionamendua hobeto ulertzeko 9. Irudian ikusten den bezala sarrerako aldagaiaren goi eta behe limiteekin proportzionaltasuna aplikatuz irteerako balio erreala kalkulatu du.

UNSCALE funtzioak kontrakoa egiten du, balio erreal bat bitarrean bihurtzen du. Funtzio honek ere hiru sarrera eta irteera bat ditu: sarrerako balio erreal, balio errealaren goi limitea, balio errealaren behe limitea eta azkenik irteerako balio bitarra.

SCALE funtzioa sentsoeren balioa neurtzeko erabili da eta UNSCALE berriz PLC-aren irteera analogikora berogailu edo haizegailura kitzikadura tentsioa bidaltzeko.

5.3. TIA PORTAL-EN SOFTWARE ARKITEKTURA

Aurretik esan bezala kontrol sistemaren diseinua egiteko Siemens-en TIA Portal V13 programa erabili da. Software hau estandarizazio altukoa da, IEC 61131-3 estandarri jarraituz.

Kontrol sistemaren arkitektura eraikitzeko estandarrak finkatzen duen bezala programa blokeak erabili dira. Programa, funtzio eta funtzio bloke hauek estandarrak POU bezala ezagutzen ditu. TIA Portal-ek lau programa bloke bereizten ditu:

- OB: Antolaketa blokeak dira eta zikloaren une zehatz batzuetan exekutatzen dira. OB mota desberdinak daude eginkizun desberdinetarako, baina lan honetan bi mota erabili dira soilik: OB1 eta OB30. OB1 programa orok behar duen bloke nagusia da, PLC martxan jartzean etengabe exekutatzen da.

Bestetik, OB30-a "Cyclic interrupt" motakoa da. Aurretik finkatutako ziklo denbora bakoitzean exekutatzen da bloke nagusia etenez.

- FC: Memoriarik gabeko bloke funtzio logikoa.
- FB: Memoriadun bloke funtzio logikoa, exekutatzen diren aldi bakoitzean aurreko exekuzioko datuak gordeta dituzte. Parametro formalak erabiliz nahi adina instantziatu daitezke.
- DB: Datuak gordetzeko erabiltzen diren blokeak dira.

5.3. PROZESUAREN EREDUA

Begizta irekiko eredua lortzeko metodologia hurrengoa da:

Lehendabizi sistema operazio-puntu batera eraman behar da. Hori lortzeko, aukeratutako potentzia batez kitzikatuko da sistema eta egonkortzen utziko da. Behin egonkortu delarik, potentzian inkrementu edo dekrementu bat sartuko da. Potentziaren aldaketa hasten den unetik berriro egonkortu arte sentsoreen temperatura neurtzen da, temperaturaren bilakaera lortuz.

Grafiko honetatik lortzen da sistemaren eredu matematikoa eta kontrolagailuaren diseinura pasa daiteke.



Irudia 10: Prozesuaren eredua lortzeko eskema.

Sistema bi eragingailuetatik kitzikatu daiteke, berogailutik edo haizegailutik, eta potentzia egokiarekin eragiteko 5.2. seinaleen tratamenduaren kontzeptuak aplikatu behar dira. Era berean egin behar da hiru sentsoreen neurketak jasotzeko

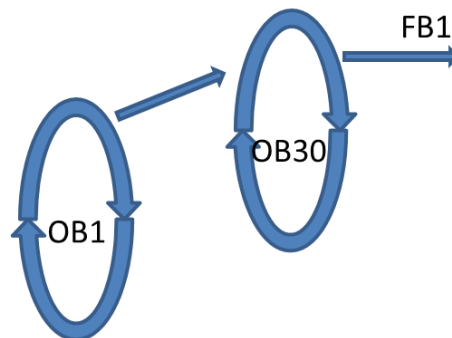
Lehen azaldu bezala, sistema bi puntutan egonkortzen utzi behar da prozesuaren eredua lortzeko. Egonkortzea programan tenporizadoreen bitartez egin da, hau da, erabiltzaileak erabakiko du zenbat denbora behar duen sistemak egonkortzeko.

5.3.1. SOFTWARE DISEINUA ETA INPLEMENTAZIOA

Software honen arkitekturarako, 11. Irudian ikusi daitekeen bezala, OB1 erabili da bloke nagusi bezala eta OB30-a ('cyclinc interrupt') aurretik ezarritako ziklo bakoitzean bloke nagusia eten egingo duen. Eten bakoitzean eraiki den FB-a exekutatu da. FB horrek berogailuaren sarrerako pulsu karratua sortuko du eta aldi berean OB30-aren ziklo bakoitzean sentsorearen neurketa egingo du.

Azkenik, tenperaturaren grafikoa lortzeko TIA Portalek eskaintzen duen **Traces** tresna erabiliko da.

FB bat erabiltzeko arrazoia parametro formalak erabiltzen dituela da, horrela nahi beste alditan instantziatu daiteke eta beste sistemetara hedatzeko aukera ematen du.

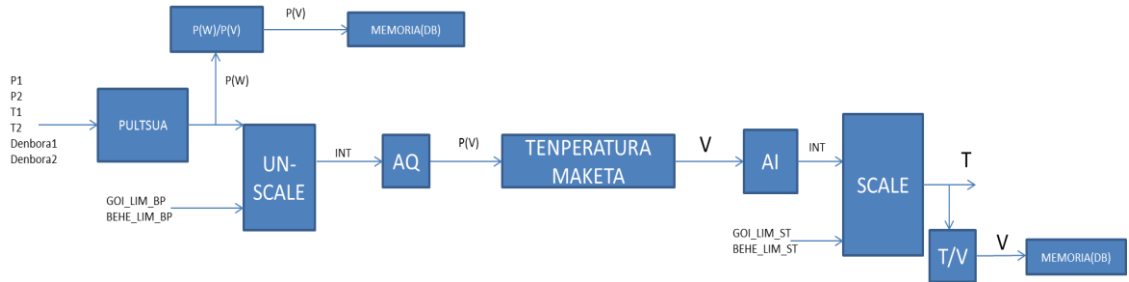


Irudia 11: Software arkitekturaren eskema.

- **FB1:** Funtzio bloke honek tenperatura maketaren berogailuaren edo haizegailuaren sarrera aukeratutako potentziarekin kitzikatzen du eta aldi berean sentsoretik datorren tenperaturaren irakurketa egiten du.

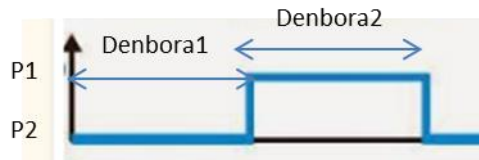
5.2. atalean azaldu den bezala PLC-aren irteeran eta sarreran seinale elektrikoak ditugu, eta PLC-aren sarrera eta irteera analogikoen moduluek seinale analogiko batera pasatzen dute. Seinale hauen identifikazioa eta tratamendua karakterizazioa deitzen da.

Kasu honetan, bi eragingailu ditugu, berogailua eta haizegailua, eta aldiz hiru sentsore; horregatik sarrera eta irteera horien karakterizazioa guztiz garrantzizkoa da FB-an. Ondorengo irudian argiago ikusiko da FB-aren funtzionamendua.



Irudia 12: FB-aren funtzionamenduaren irudi eskematikoa.

Lehenik eta behin, programak P1, P2, T1, T2, Denbora1, Denbora2 eta Martxa sarrera aldagaiekin pultsu karratu bat sortzen du. P1 eta P2 pultsu karratuaren behe eta goi balioak dira hurrenez hurren, eta REAL motako aldagai hauek gisa definitu dira, beraien unitateak watt-etan emanez; T1 eta T2 pultsua sortzeko beharrezkoak diren tenporizadoreak dira; Denbora1 eta Denbora2 tenporizadore horien denborak dira, eta azkenik, pulsuari hasiera emateko 'martxa' (%10.0) sarrera eragingo dugu. 13. Irudian ikusten da sarrerako pultsu karratua.



Irudia 13: Eragingailuaren sarrerako pulsu karratua.

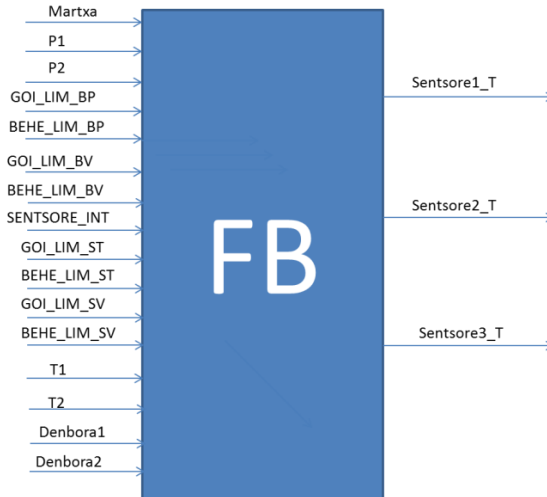
Aurreko eragiketa egin ondoren, watt-etan dugun sarrerako potentzia hau INTEGER (bitarra) formatura pasatzen da, eragingailuaren sarrerara bideratzeko seinale bihurtzeko. Horretarako UNSCALE funtzioa erabili da. Beraz, BEHE_LIM_BP eta GOI_LIM_BP berogailuaren balio minimoa eta maximoa izango direnak sartu behar dira honen karakterizazioarako.

Bestetik, FB-ak sentsoreek ematen duten temperatura neurtu behar du. Horretarako, PLC-aren sarrera modulan neurtzen den balio bitarrak REAL formatura pasa behar dira, SCALE erabiliz.

Hurrengo orrialdeko 2. Taulan, FB-aren sarrera eta irteeren azalpena dago.

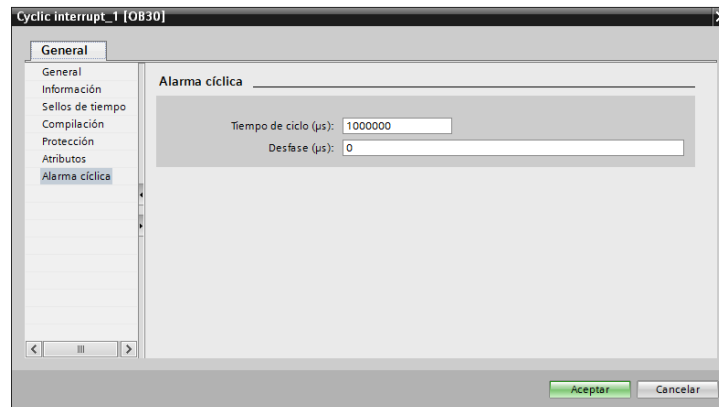
INPUT		
MARTXA	Berogailuaren pulsuari hasiera ematen dion marka	BOOL
P1	Berogailuaren pulsuaren beheko balioa, watt-etan emanda	REAL
P2	Berogailuaren pulsuaren goiko balioa, watt-etan emanda	REAL
GOI_LIM_BP	Potentziaren karakterizazioa egiteko potentziaren balio maximoa 27648 balio bitarrari dagokiona, watt-etan emanda	REAL
BEHE_LIM_BP	Potentziaren karakterizazioa egiteko potentziaren balio minimoa 0 balio bitarrari dagokiona, watt-etan emanda	REAL
GOI_LIM_BV	Potentziaren karakterizazioa egiteko potentziaren balio maximoa 27648 balio bitarrari dagokiona, V-etan emanda	REAL
BEHE_LIM_BV	Potentziaren karakterizazioa egiteko potentziaren balio minimoa 0 balio bitarrari dagokiona, V-etan emanda	REAL
SENTORE1_INT SENTORE2_INT SENTORE3_INT	Sentsoreen sarrera analogikoak dira, PLC-aren sarrerara konektatzen direnak	INT
GOI_LIM_ST	Sentsorearen tenperaturaren karakterizazioa egiteko sentsoreak sumatu dezakeen tenperatura maximoaren balio 27648 balio bitarrari dagokiona, °C-etan emanda	REAL
BEHE_LIM_S	Sentsorearen tenperaturaren karakterizazioa egiteko sentsoreak sumatu dezakeen tenperatura minimoa balio 0 balio bitarrari dagokiona, °C-etan emanda	REAL
GOI_LIM_SV	Sentsorearen tenperaturaren karakterizazioa egiteko sentsoreak ematen duen boltaje maximoa 27648 balio bitarrari dagokiona, V-etan emanda	REAL
BEHE_LIM_SV	Sentsorearen tenperaturaren karakterizazioa egiteko sentsoreak ematen duen boltaje minimoa 0 balio bitarrari dagokiona, V-etan emanda	REAL
T1 T2	Pultsua sortzeko beharrezkoa den tenporizadoreak	TIMER
Denbora1 Denbora2	Pultsua definituko duen denbora tarteak	S5TIME
OUTPUT		
Sentsore1_T Sentsore2_T Sentsore3_T	Sentsoreak sumatzen duen tenperatura °C-tan emanda	REAL

Taula 2: Begizta Irekiko FB-aren sarreren eta irteeren taula



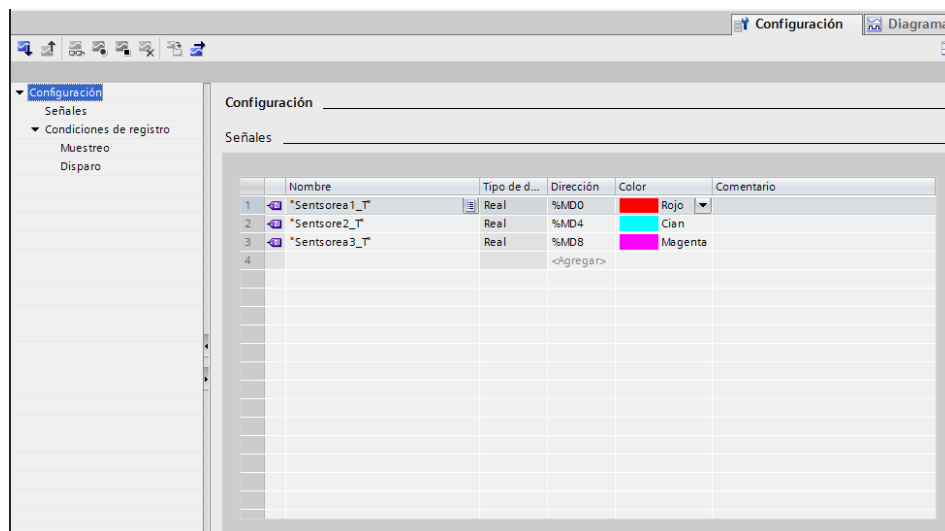
Irudia 14: Eredua lortzeko programaren FB-a

- OB30:** Aurretik esan bezala FB-a ziklikoki exekutatzeko OB30-etik egin da funtzio blokearen deia. Honek aurretik ezarritako ziklo bakoitzean programa nagusia geldituko du eta bertan FB exekutatuko da. Ziklo denbora aukeratzeko sistemaren dinamika ezagutu behar da. Prozesua oso azkarra bada ez du zentzurik ziklo denbora handia jartzea, tarteko neurketa asko galtzen baitira. Era berean, prozesu geldo batean, ziklo-denbora txikia jarri ezker, beharrezkoak baino neurketa gehiago hartuko dira eta PLC-aren memoria alferrik beteko da. Tenperaturaren kasuan, oso dinamika geldoko prozesua da beste sistema batzuekin konparatuta, adibidez motor baten dinamikarekin. Modu enpirikoan, esperimentuen bitartez, erabaki da 1 segundoko ziklo-denbora ezartzea.



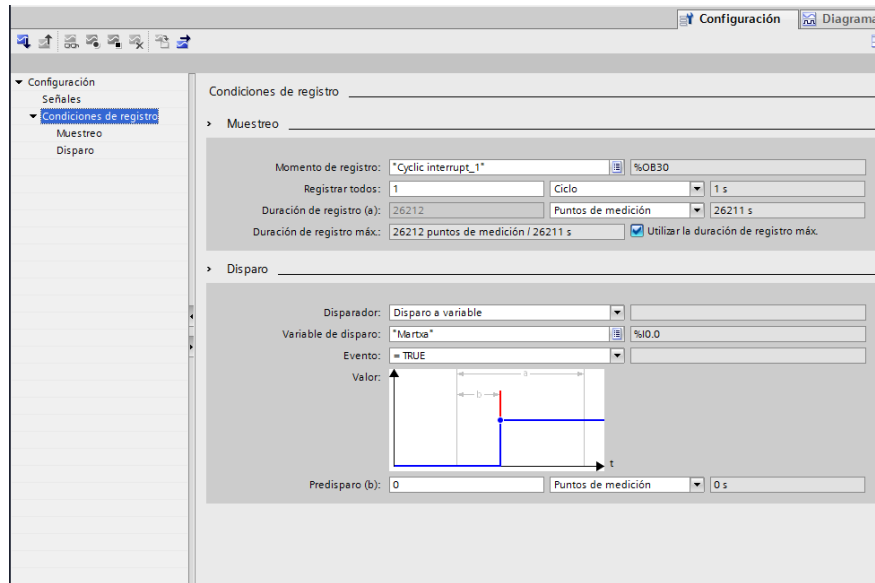
Irudia 15: Prozesuaren irudiaren OB30-aren konfigurazioa.

- Traces:** Ereduaren irudi grafikoa lortzeko, TIA Portal-ek bere azkeneko bertsioetan eskaintzen duen Traces komandoa erabili da. Erreminta honek aldagai bat baino gehiagoren grafikoa aldi berean erregistratu eta erakusteko aukera ematen du. Memoria murriztua dauka, beraz, dinamikaren azterketa eginez, lagin-kopuruaren aukeraketa egokia egin behar da memoriaren optimizaziorako. Sortu den Traces konfigurazioan, OB30-ari lotu da eta aurretik azaldu den ziklo bakoitzeko Traces-ek sentsoreen neurri bana hartzen du. 15. eta 16. Irudietan ikus daiteke konfigurazioa.



Irudia 16: Prozesuaren irudiaren Traces konfigurazioa.

17. irudian ikusten den bazala neurketen erregistroa "Martxa" (%10.0) sarrera eragitean hasiko da.



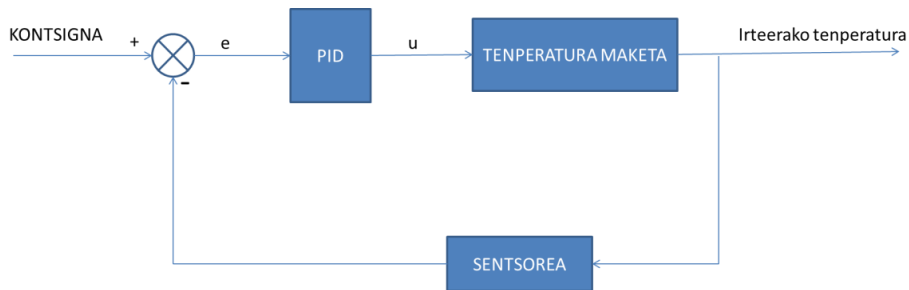
Irudia 17: Prozesuaren irudiaren Traces konfigurazioa.

5.4. KONTROL ZUZENA

Begizta irekiko esperimentuarekin lortutako grafikotik, prozesuaren eredua lortu ahal izango da. Eredua horretaz baliatuz, PID-aren parametroak kalkulatu ahal izango dira begizta itxiko kontrolean erabili ahal izango direnak.

Programaren zati honetan, PID-aren inplementazioa egingo da. Horretarako, PID ez iteratiboa erabiltzea erabaki da, (5.1.3. atalean azaldu dira PID honen ezaugarriak).

Kontrol-sistema bat eraiki nahi dugunean, 16. irudian ikusten den eskema jarraitu behar da. Demagun prozesua operazio-puntu batean dagoela egonkorturik. Operazio-puntuko tenperatura eta konsignako edo erreferentziako tenperatura berdinak badira, sistemaren errorea zero izango da. Erreferentzian aldaketa bat sartzen dugun unean, gorakorra edo beherakorra izan daitekeena, errore bat izango dugu, konsignako balioa ez baita izango sentsoreak ematen duen balio berdina. Errorea dagoen unetik aurrera, PID-ak etengabe, laginketa-ziklo bakoitzean, errorea detektatu eta kontrol-seinale edo akzioa kalkulatu du, eragingailuan (berogailura) kitzikatu beharreko potentzia kalkulatu, berriro ere errorea nulua izan arte.



Irudia 18: Kontrol-sistema baten eskema.

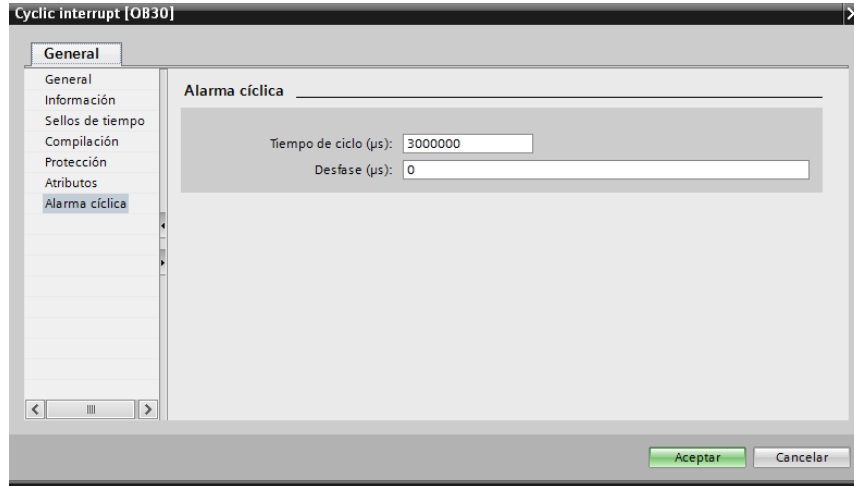
5.4.1. SOFTWARE DISEINUA ETA INPLEMENTAZIOA

Programa honen software arkitektura aurreko programaren antzekoa da: OB1 "Main" dugu bloke nagusi bezala, ziklikoki exekutatu den OB30-a eta azkenik FB bat dugu dagokion DB-arekin. Berriro ere, emaitza grafikoak lortzeko Traces erabiliko da.

PID								
	Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible d...	Visible en ..	Valor de a..	Comentario
1	▼ Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	▣ Kc	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	▣ T	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	▣ Tí	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	▣ Td	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	▣ N	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	▣ sentsorea_int	Int	0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	▣ setpoint_T	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	▣ GOI_LIM_ST	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	▣ BEHE_LIM_ST	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	▣ GOI_LIM_SV	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	▣ BEHE_LIM_SV	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	▣ OP_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	▣ AUTOMATIKO	Bool	false	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	▼ Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
16	▣ u_analog	Int	0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17	▣ sentsore_T	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
18	▣ u	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19	▼ InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
20	▣ <Agrega>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
21	▼ Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
22	▣ e	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
23	▣ up	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
24	▣ ui	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
25	▣ ui1	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
26	▣ e1	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
27	▣ ud	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
28	▣ ud1	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
29	▣ error1	Word	16#0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
30	▣ error2	Word	16#0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
31	▣ sentsore_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
32	▣ setpoint_V	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
33	▣ u_W	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
34	▣ Karakterizazio_faktora	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
35	▣ GOI_LIM_ST_karak	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
36	▣ GOI_LIM_SV_karak	Real	0.0	No remane...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
37	▼ Temp				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
38	▣ <Agrega>				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
39	▼ Constant				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

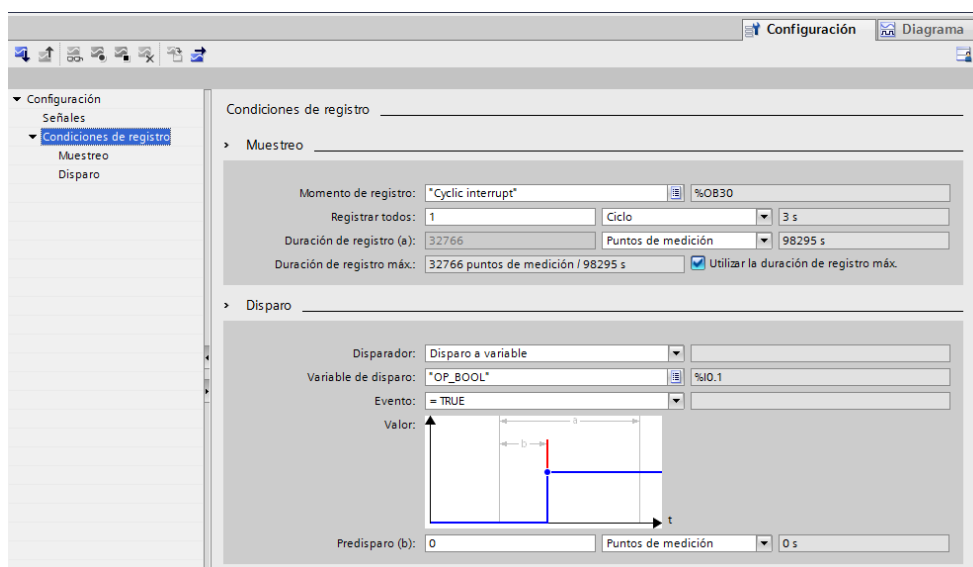
Irudia 20: Kontrol zuzenaren FB-aren sarrerak eta irteerak.

- OB30:** Lehen bezala, ziklo-denbora OB30-aren konfigurazioaren bitartez ezarriko da. Kasu honetan, ziklo-denborak PID-aren laginketa-tartea finkatzen du, eta honek PID-ren funtzionamenduan eragiten du. Berrito ere, laginketa-tartea esperientziatik ateratako konklusioetatik ondorioztatu da. Temperaturaren dinamikari erreparatuz, 3segundoko periodoa ezarri da.



Irudia 21: Kontrol zuzeneko OB30-aren konfigurazioa.

- Traces:** Aldagaiak grafikatzeko sistema aurrekoaren oso antzekoa da. Kasu honetan bi grafiko sortuko dira Traces-en bidez. Alde batetik, Setpoint-a edo konsigna sentsoretik jasotako tenperaturaren bilakaerarekin konparatuko da eta bestetik, PID-ak kitzikatzen duen tentsioa erakutsiko da. Traces-ek ziklo bakoitzean egingo ditu neurketak eta "AUTOMATIKO" (%I0.1) sarrera eragitean hasiko da neurketen erregistroarekin. Hurrengo irudietan Traces-en konfigurazioa erakusten da:



Irudia 22: Kontrol zuzeneko Traces-aren konfigurazioa.

6. APLIKAZIOAREN ADIBIDEA

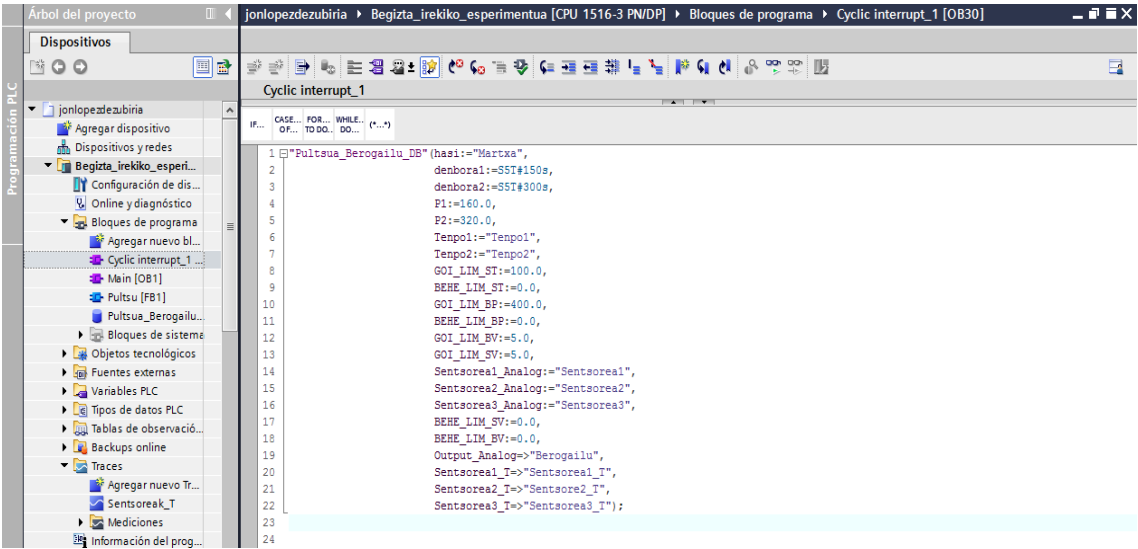
Atala honen helburua proiektuaren aplikazio praktikoa erakustea da software honen erabiltzaileei lagungarria egingo zaiolakoan.

Atal honek hiru azpiatal izango ditu kontrol-sistema bat eraikitzeko pausoekin elkartuta daudenak. Lehenik, begizta irekiko esperimentuaren bidez aurretik aukeratu den operazio-puntuaren sistemaren eredu lortuko da, ondoren lortutako grafikotik transferentzia-funtzioa lortu eta PID kontrolagailua diseinatuko da, eta azkenik, egindako kontrolaren diseinua era praktikoa egiaztatuko da.

6.1. PROZESUAREN EREDUA LORTZEKO ESPERIMENTUA

Esperimentu honetan, hasieran sistema 160 W-ekin kitzikatu egin da 150 segundoz. Denbora horretan sistema egonkortzeko adina denbora duela estimatu da. Behin denbora hori igarota, berogailuaren sarrera beste 160 W inkrementatu da, hau da 320W izango ditugu berogailuaren sarreran. 300 segundo horietan, sistema egonkortuko da eta ondoren berriro ere hasierako operazio-puntura itzuliko da.

Sarrerako pultsuri hasiera emateko, "Martxa" (%I0.0) sarrera sakatu behar da.

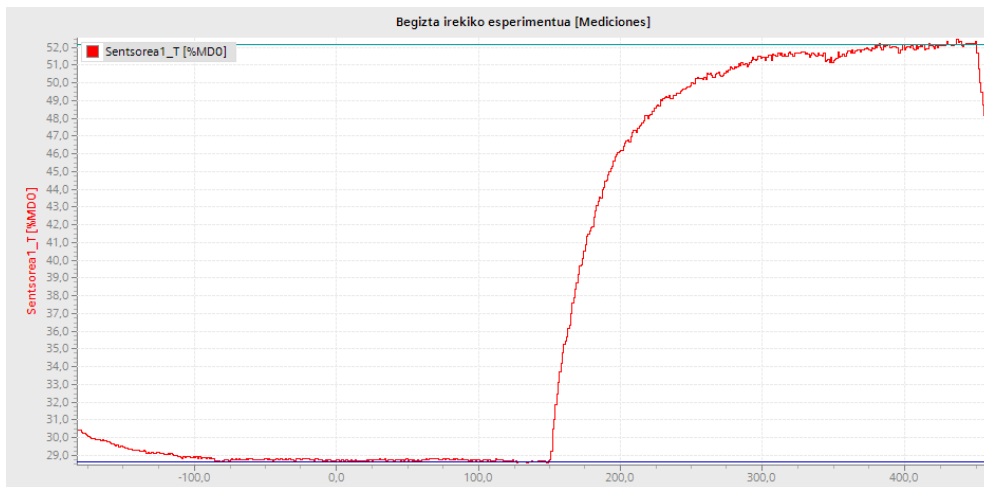


```

1  "Pultsua_Berogailu_DB" (hasi:="Martxa",
2      denbora1:=SSI#150s,
3      denbora2:=SSI#300s,
4      F1:=160.0,
5      F2:=320.0,
6      Tempo1:="Tempo1",
7      Tempo2:="Tempo2",
8      GOI_LIM_ST:=100.0,
9      BEHE_LIM_ST:=0.0,
10     GOI_LIM_BP:=400.0,
11     BEHE_LIM_BP:=0.0,
12     GOI_LIM_SV:=5.0,
13     GOI_LIM_SV:=5.0,
14     Sentsorea1_Analog:="Sentsorea1",
15     Sentsorea2_Analog:="Sentsorea2",
16     Sentsorea3_Analog:="Sentsorea3",
17     BEHE_LIM_SV:=0.0,
18     BEHE_LIM_SV:=0.0,
19     Output_Analog->"Berogailu",
20     Sentsorea1_I->"Sentsorea1_I",
21     Sentsorea2_I->"Sentsorea2_I",
22     Sentsorea3_I->"Sentsorea3_I");
23
24
  
```

Irudia 23: OB30-etik egindako instantzia,

Balio hauek, aldagaien karakterizazioarentzat beharrezkoak diren balioekin batera, OB30-en egindako **Pultsu_Berogailu_DB**-ra egindako instantzian sartu behar dira. Aldagaien karakterizaziorako behar diren datuak 5.1. atalean maketaren deskripzioan eman diren datuetan daude. 24. irudian lortutako grafikoa ikusi dezakegu.



Irudia 24: Traces-eki lortutako 1.go sentsorearen grafikoa. Y ardatzean temperatura °C-tan eta X ardatzean denbora segundotan.

6.2. KONTROLAGAILUAREN DISEINUA

Behin eredua dugula, kontrolagailua diseinatu behar da. Lehen pausoa iruditik hurbilpen matematikoa lortzea da. Ikusten den bezala sistema lehen ordenako sistema baten erantzunari hurbiltzen zaio, beraz, ondorengo eredu bidez adieraz dezakegu:

$$G(s) = \frac{K}{1 + \tau s} = \frac{T(s)}{P(s)}$$

K eta τ irabazpena eta denbora-konstantea dira, eredua definitzen duten parametroak, eta grafikotik kalkula daitezkeenak:

$$K = \frac{\Delta T}{\Delta U} = \frac{52,72^{\circ}\text{C} - 36,13^{\circ}\text{C}}{4\text{V} - 2\text{V}} = 8,295^{\circ}\text{C}/\text{V}$$

$$0,632\Delta T - T_0 = 0,632 \times 16,59^\circ\text{C} + 36,13^\circ\text{C} = 46,615^\circ\text{C}$$

$$\text{Temperatura horri dagokion denbora} \rightarrow \tau = 36,762s$$

Sistemaren transferentzia funtzioa orduan: $G(s) = \frac{8,295}{1+36,762s}$

Sistema identifikatu eta gero kontrolagailuaren sintonizazio analitikoa egin da. Lehen ordenako sistema bat dugu eta helburua da errore estatiko nulua izatea, hau da sistema 50 °C-tan egonkortzea. Horretarako, akzio integralaren beharra dago. Adibide honetan akzio diferentziala alde batera utziko da, beraz PI bat diseinatuko da.

$$U(s) = Kc \frac{1 + Tis}{Tis}$$

PI-aren ekuazioa aurrekoa izanik, kontrolagailuaren zeroarekin sistemaren poloa ezabatuko da. $Ti=36,762$.

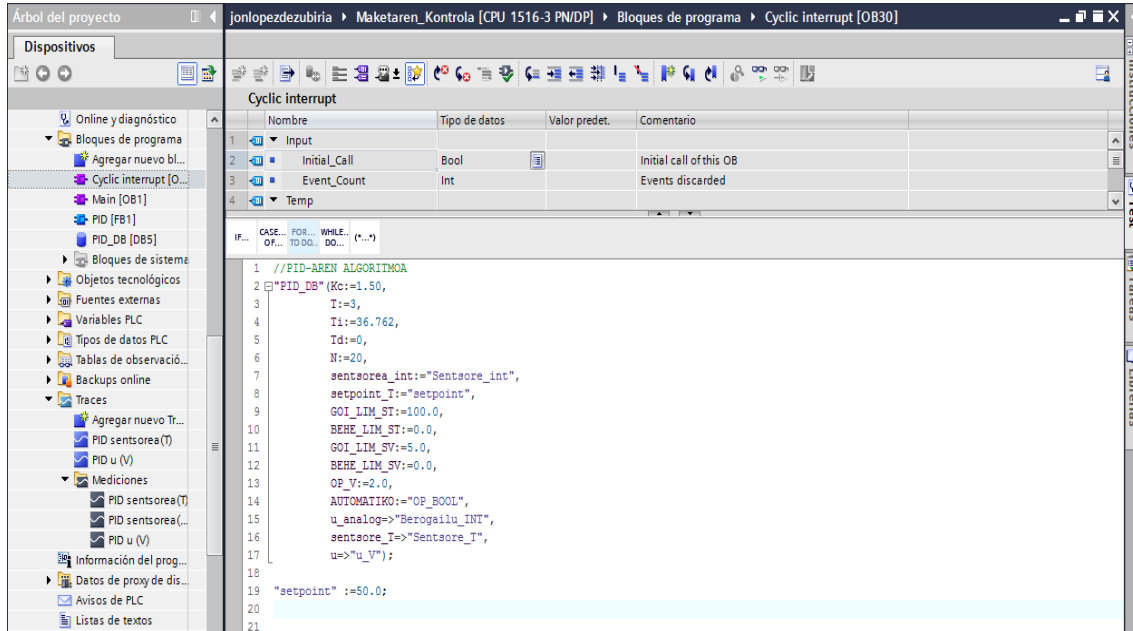
Kc -rekin sistemaren dinamika azkartuko da. Sistemaren dinamika ezin da nahi adina azkartu eragingailuak mugatua baitago, ezin du 5 V baino gehiago eman. Beraz, Kc -ren aukeraketarako irizpidea eragingailua ez asetzea da. Asetasuna egotekotan hasierako unean izango da, errora maximoa denean.

Kc -ri 1,5-eko balioa eman zaio.

Hona hemen diseinatutako PI kontrolagailua: $Gc(s) = 1,5 \frac{1+36,762s}{36,762s}$

6.3. KONTROLAGAILUAREN BALIZKOTZEA

Kontrolagailuaren diseinua eginik, kontrolaren funtzionamendua egiaztatuko da. Horretarako, lehenbizi aurreko operazio puntura eramango da sistema 160W-rekin kitzikatuz. Behin sistema egonkortu denean "AUTOMATIKO" (%10.1) sarrera eragingo da PI kontrolari hasiera emateko.



Irudia 25: Kontrol zuzeneko OB30-ari eginiko instantzia.

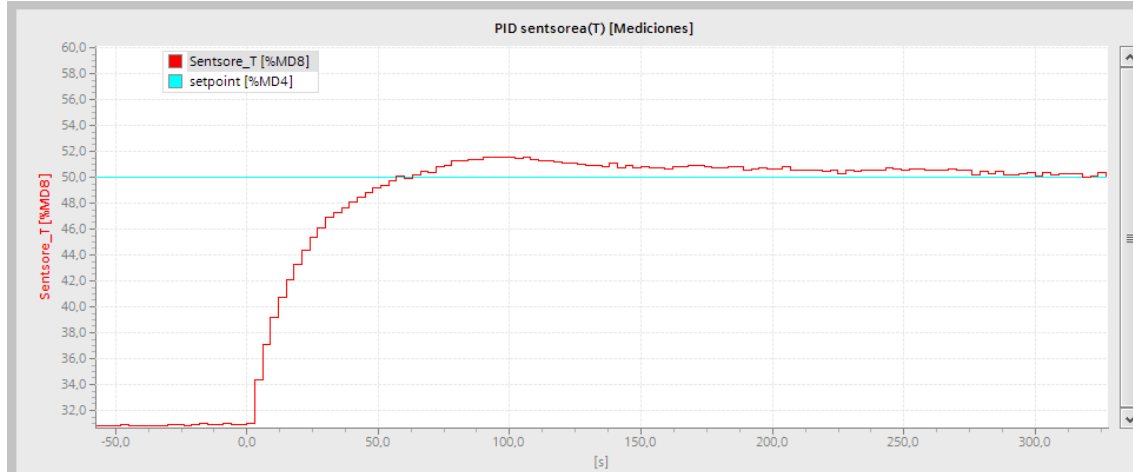
Instantzia honetan, 24. Irudia ikusten den bezala, PID-aren parametroez gain sentsorea eta berogailuaren karakterizazioarako datuak sartu behar dira, begizta irekiko esperimentuan egin den bezala.

Kontrolagailua diskretizatzerakoan, T aldagai bat sortu dugu, laginketa-tartea dena. 6.4.2. "Begizta itxiko kontrolaren konfigurazioa" atalean ikusi den bezala, laginketa-tartea 3 segundokoa ezarri da.

Programaren funtzionamendu egokirako konexioei dagokionez:

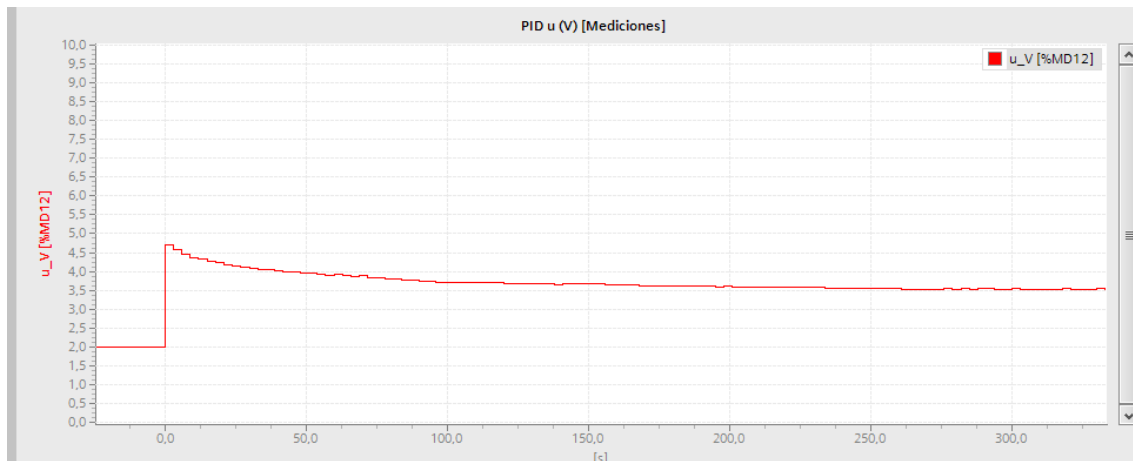
- **Berogailu_INT:** %QW4 maketaren berogailuaren sarrerara.
- **Sentsore_T:** %IW4 maketaren 1.go sentsorearen irteerara.

Programa PLC-an kargatu eta on-line konexioa aktibatuta dagoela, "AUTOMATIKO" (%I0.1) sarrera eragiterakoan Traces bidezko erregistroa hasiko da.



Irudia 26: Traces-en neurketa, Sentsorea eta Setpoint-a, temperatura °C-etan denboraren funtzioan segundotan.

25. irudian ikusten den bezala tenperaturak erreferentziari jarraitzen dio. Tenperaren dinamika oso motela da horregatik egonkortzeko 300segundo inguruan behar ditu.



Irudia 27: Traces-en neurketa, berogailuaren kitzikadura denborarekiko, tentsioa V-etan denbora segundotan.

Lehen azaldu bezala errore maximoa kontrola hasten den momentuan ematen da, beraz une horretan berogailuan sartzen den kitzikadura maximoa da. Ondoren pixkanaka jaisten da egonkortu arte. 26. Irudian ikusten den bezala eragingailuaren saturazio arriskua hasierako unean dago, baina kasu honetan ez

da saturaziorik gertatzen berogailuaren kitzikadura maximoa 4,7V ingurukoa da berogailuaren limitetik behera.

6.4. ESPERIMENTUAREN ANALISIA

Alde batetik, esperimentu honekin egiaztatu da software honen erabilgarritasuna, kontrol-sistema bat inplementatu baita tenperatura maketan.

Temperaturak oso dinamika geldoa dauka beraz egonkortzeko denbora asko behar du, beraz prozesuaren eredia lortzeko egonkortze-denbora altuak jartzea gomendagarria da.

Eragingailuaren mugapena ere kontuan hartzekoa da, maketaren deskribapenean ikus daitekeen legez: tenperaturak har dezakeen tenperatura maximoa 60°C ingurukoa da, eragingailuaren eragiteko tartea 0-5 V artekoa eta linealtasun tartea 2-5 V, beraz, operazio-puntuaren arabera, eragingailua asetzeko arriskua dago. Horregatik, kontrolagailuaren Kc aukeratzekoan aurrekoari erreparatu behar zaio.

7. BETEBEHARREN DESKRIBAPENA

Gradu amaierako lana 2017-2018 ikasturtean zehar egin da, zehatzagoak izateko urtarrilaren 7tik uztailaren 8rarte. Denbora-bitarte horretan, arrozoi pertsonal eta akademikoengatik geldialdiak kontuan hartu behar dira ere.

Egun bakoitza bataz beste 4 orduko lan ordu bezala kontsideratu da.

Lanaren antolaketa 4 tarteetan zatitu da: Lanerako tresnen ezagutza, alternatiben aukeraketa, Programazioa, Balidazioa eta Dokumentazioa.

1.Lanerako tresnen ezagutza: PLC-ak aurretik esan bezala potentzial handiko tresnak dira konplexutasun handiarekin. Lan honetan oso tresna potentea erabili da betebegar sinpleago bat betetzeko, beraz aurreko pausoa tresneriaren ezagutza da.

Ezagutza hau esperimentu, froga-erroreen eta dokumentazioaren bitartez lortu da. Hiru erronka nagusirekin aurkitu gara: tratamendu analogikoa, SCL programazio lengoaiaren ezagutza eta Traces-en erabilera.

Erronka hauek aurrera eramateko denbora 15 egunekoa izan da.

2. Alternatiben aukeraketa: Maketaren kontrol-sistema erabakitzea. Egun batean egin da.

3. Programazioa: zati honetan bi FB eraiki dira, bata begizta irekiko kontrola lortzeko eta bestea sistemaren kontrola lortzeko. Aipatu beharra dago FB hauen programazioa ez dela berehalakoa izan lehen aipatutako konplexutasunagatik froga errore ugari izan dira.

Guztira beste 15 eguneko iraupena eman zaio atal honi.

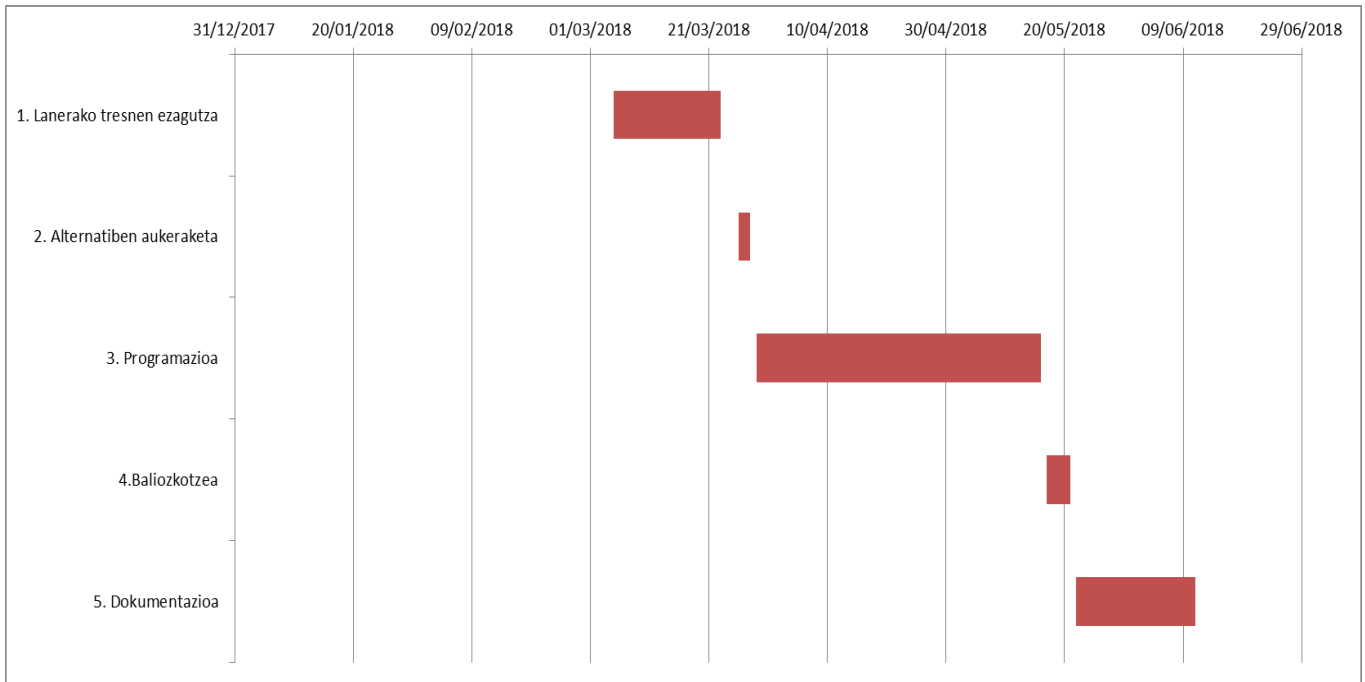
4. Baliozkotzea: Behin programa eginda dagoela, funtzionatzen duela frogatu egin da esperimentu ezberdinak eginik. Esperimentu hauek sistemaren operazio puntu desberdinetan egin dira.

Betebegar honetan 3 egun inbertitu dira.

5. Dokumentazioa: Bertan gradu amaierako lana denaren antolaketa eta idazketa barne sartzen dira. Dokumentu hau idazten 15egun erabili dira.

Guztira Gradu Amaierako Lan honetan 49 egun inbertitu dira, $49 \times 4h = 196h$ suposatzen dutenak.

7.1. GANTT DIAGRAMA



Irudia 28: Betebeharren Gantt diagrama

8. AURREKONTUA

Proiektu honetarako gastuak atal desberdinetan banatu dira:

- **Barne orduak:** Bertan proiektu honetan inbertitu den denbora kontuan hartu da. Alde batetik Ingeniari Junior-aren figura sartu da, Industria Teknologien Ingeniaritzan graduatu, bertan Gantt diagraman agertzen diren proiektua aurrera eramateko betebeharrak nagusiak kontuan hartu dira. Bestetik, Ingeniari Seniorraren figura dago, proiektuaren zuzendaria dena. Azken honen betebeharrak proiektua zuzentzea eta gidatzea da.
- **Gastuak:** Atal honetan batez ere bulegoko materiala sartu da: koadernoak, boligrafoak, fotokopiak ...
- **Amortizazioak:** Bertan proiektu honetan erabilerarengatik tresneriak izan duen balioaren galera kontuan hartu da. Alde batetik, hardwarea: SIMATIC S7-1500, Quanser Heat Flow Experiment eta ordenagailua. Bestetik, softwarea: TIA Portal V13.
- **Zeharkako Kostuak:** Atal honetan zeharkako gastuak kontuan hartu dira: elektrizitatea, Internet, laborategiaren garbiketa ... Kostu zuzenen %3 direla estimatu da.
- **Ezusteak:** Ezusteak aurreikusten den diru kopurua da. Bertan, ekipoen matxuren konponketa eta beste edonolako ezbeharrak kontuan hartzen dira.

BARNE ORDUAK				5.650 €
KONTZEPTUA	UNITATEA	UNITATE KOPURUA	KOSTU UNITARIOA	KOSTUA
INGENERI JUNIORRA	H	196	25 €	4.900 €
INGENERI SENIORRA	H	15	50 €	750 €
GASTUAK				15 €
KONTZEPTUA	UNITATEA	UNITATE KOPURUA	KOSTU UNITARIOA	KOSTUA
BULEGOKO MATERIALA				15 €
AMORTIZAZIOAK				965,90 €
KONTZEPTUA	UNITATEA	UNITATE KOPURUA	KOSTU UNITARIOA	KOSTUA
QUANSER HEAT FLOW ESPERIMENT	PORTZENTAIA	5%	250 €	13 €
PLC	PORTZENTAIA	5%	3.500 €	175 €
ORDENAGAILUA	H	196	0,40 €	78,40 €
Lizentzia TIA Portal				700,00 €
SUBTOTALA1				6.630,90 €
Zeharkako Kostuak(3%)				198,927
SUBTOTALA2				6.829,83 €
Ezusteak(%10)				682,98 €
GUZTIRA				7.512,81 €

Taula 3: Proiektuaren aurrekontua.

Esanguratsua da aurrekontuko zatirik handiena barne orduak direla. Tresneria profesional batekin lan egin denez normala da graduko ingeniari batek denbora asko inbertitzea makinaren ezagutzan eta proiektuaren gauzatzean.

9. ONDORIOAK

Proiektu honetan tenperatura maketaren identifikaziorako eta kontrolerako software bat eraikitzea lortu da, 6. atalean “Aplikazioaren adibidea” egiaztatu den bezala. Horretarako, SIMATIC S7-1500 PLC-a eta TIA Portal programa erabili dira. Kontrol sistema baten diseinuak hiru fase dituzte: sistemaren eredu lortzea, kontrolagailuaren aukeraketa eta sintonizazioa eta kontrol zuzena, hauek ahalbidetzea izan da programaren helburua.

Prozesuaren identifikaziorako softwarean, berogailuan edo haizegailuan potentzia diferentzia bat sortzen duen eta sentsoaren seinaleak erregistratzen dituen programa bat sortu da. Lortutako emaitza hauek prozesuaren eredu matematiko bat emango digu. Eredu matematikoak oso baliagarriak dira kontrolagailuaren aukeraketa eta parametroen sintonizazioa egiteko.

Kontrol zuzena inplementatzeko programan, PID ez interatiboaren (ideala) algoritmoa diskretizatu eta programatu da. Beste edozein PID bat erraz inplementatu programaren barruan sartuz eta PID-aren kodigoa aldatuz.

Programa eraikitzerakoan helburuetako bat kodigoaren berrerabilpena izan da, horregatik programaren arkitekturan bloke funtzionalak erabili dira nahi adina aldiz parametro formalekin instantziatu daitezkeenak.

Bestetik, kontuan hartzekoa da ere, hainbatetan errepikatu den tresneriaren profesionaltasuna, aplikazio honentzako potentzialtasun osoaren zati txiki bat erabili da besterik ez. Hori dela eta programa eraikitzean arazo ugari suertatu dira. Hauek froga errorearen eta zeharkako esperimenduak eginez ebatzi dira, pixkanaka tresneriari buruzko ezagutzak bereganatzen.

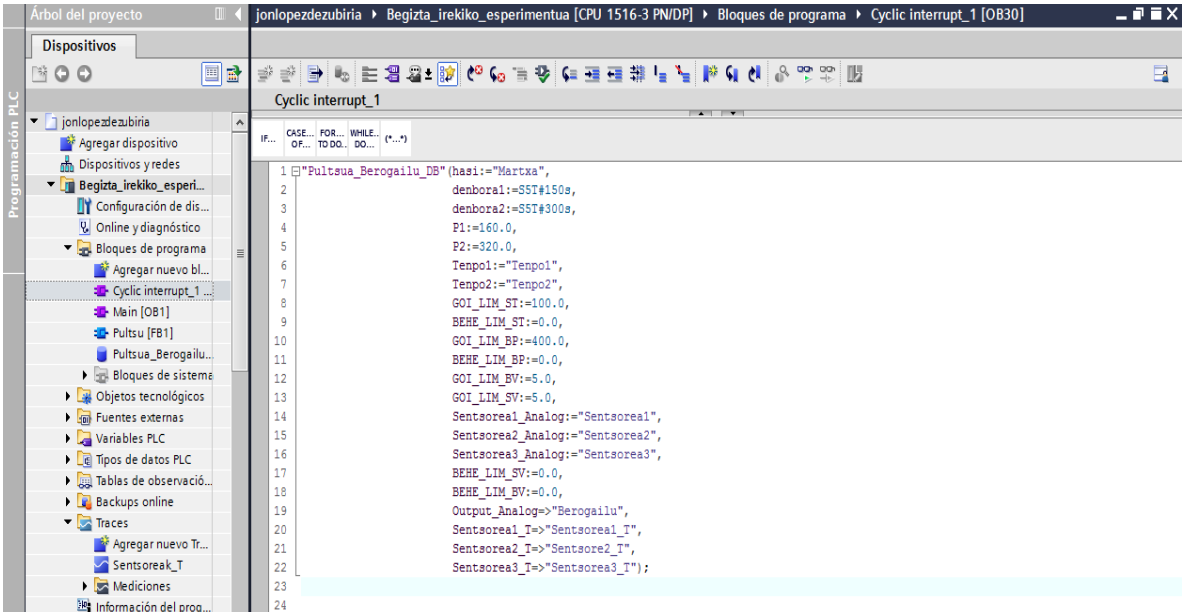
Nahiz eta programaren funtzionamendu egokia egiaztatu den, kodigoaren efizientzia eta programaren beste atal batzuk hobetu daitezkeela esan beharra dago. Proiektu hau beste baten abiapuntua izan daiteke.

INFORMAZIO ITURRIAK

- [1] Orive, D., y López, F. (2015). Automatización Industrial. Bilbo, Espainia.
- [2] Iriondo N., Marcos, M., Cabanes I., Zubizarreta A. (2014). Automatika eta Kontrola. Bilbao, España.
- [3] Karl-Heinz John, Michael Tiegelkamp. (2010). Programming Industrial Automation Systems, Alemania.
- [4] Hans Berger (2014) Automating with SIMATIC S7-1500, Configuring, Programming and Testing with STEP 7 Professional
- [5] Åström, K.J. & Hägglund, T., 2006. Advanced PID Control. 1st ed. USA: Instrument Society of America.
- [6] International Electrotechnical Commission, IEC International Standard IEC 1131-3 Programmable Controllers, Part 3: Programming Languages, 2003.
- [7] Karl-Heinz J. & Tiegelkamp, M., 2001. IEC 61131-3: Programming Industrial Automation Systems, Berlin Heidelberg: Springer-Verlag.

ERANSKINAK I: IDENTIFIKAZIORAKO PROGRAMAREN KODIGOA

OB30:



```

1 IF... CASE... FOR... WHILE... (*...*)
2 OF... TO DO... DO...
3
4 1 "Pultsua_Berogailu_DB"(hasi:="Martxa",
5     denbora1:=SST#150s,
6     denbora2:=SST#300s,
7     P1:=160.0,
8     P2:=320.0,
9     Tempo1:="Tempo1",
10    Tempo2:="Tempo2",
11    GOI_LIM_ST:=100.0,
12    BEHE_LIM_ST:=0.0,
13    GOI_LIM_BP:=400.0,
14    BEHE_LIM_BP:=0.0,
15    GOI_LIM_BV:=5.0,
16    GOI_LIM_SV:=5.0,
17    Sentsorea1_Analog:="Sentsorea1",
18    Sentsorea2_Analog:="Sentsorea2",
19    Sentsorea3_Analog:="Sentsorea3",
20    BEHE_LIM_SV:=0.0,
21    BEHE_LIM_BV:=0.0,
22    Output_Analog=>"Berogailu",
23    Sentsorea1_T=>"Sentsorea1_T",
24    Sentsorea2_T=>"Sentsorea2_T",
25    Sentsorea3_T=>"Sentsorea3_T");
  
```

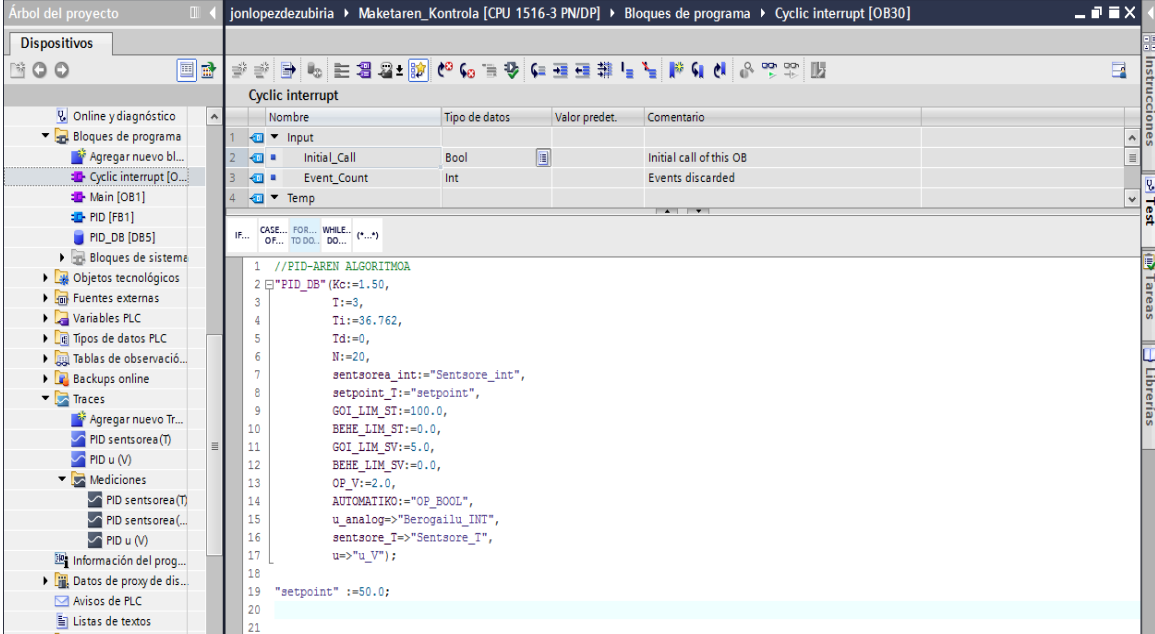
FB1:

```

1 //PULISU BOOL
2 #Tenprizadoreal:=S_ODTS(T_NO:=#Tempo1, S:=#hasi,R:=#off, TV:=#denbora1, Q=>#inkremento);
3 #Tenporizadoreal:=S_ODT(T_NO := #Tempo2, S := #inkremento, TV := #denbora2, Q => #off);
4
5 //PULISU REAL
6 IF #inkremento=0 THEN
7   #Output_P := #P1;
8 ELSE
9   #Output_P := #P2;
10
11 ;
12 END_IF;
13 //KONBERTSIOAK
14 #Karakterizazio_faktorea := 10.0 / #GOI_LIM_SV;
15 #GOI_LIM_BP_karak := #Karakterizazio_faktorea * #GOI_LIM_BP;
16 #GOI_LIM_ST_karak := #Karakterizazio_faktorea * #GOI_LIM_ST;
17 #Output_V := #Output_P * #GOI_LIM_BV / #GOI_LIM_BP+#BEHE_LIM_BV;
18
19 #error1:=UNSCALE(IN:=#Output_P, HI_LIM:=#GOI_LIM_BP_karak, LO_LIM:=#BEHE_LIM_BP, BIPOLAR:=FALSE, OUT=>#Output_Analog);
20
21 #error2 := SCALE(IN := #Sentsorea1_Analog, HI_LIM :=#GOI_LIM_ST_karak, LO_LIM :=#BEHE_LIM_ST, BIPOLAR := FALSE, OUT => #Sentsorea1_T);
22 #Sentsorea1_V := #Sentsorea1_T * #GOI_LIM_SV / #GOI_LIM_ST+#BEHE_LIM_SV;
23
24 #error3 := SCALE(IN := #Sentsorea2_Analog, HI_LIM := #GOI_LIM_ST_karak, LO_LIM := #BEHE_LIM_ST, BIPOLAR := FALSE, OUT => #Sentsorea2_T);
25 #Sentsorea2_V := #Sentsorea2_T * #GOI_LIM_SV / #GOI_LIM_ST+#BEHE_LIM_SV;
26
27 #error4 := SCALE(IN := #Sentsorea3_Analog, HI_LIM := #GOI_LIM_ST_karak, LO_LIM := #BEHE_LIM_ST, BIPOLAR := FALSE, OUT => #Sentsorea3_T);
28 #Sentsorea3_V := #Sentsorea3_T * #GOI_LIM_SV / #GOI_LIM_ST+#BEHE_LIM_SV;
29
  
```

ERANSKINAK II: BEGIZTA ITXIKO KONTROLAREN KODIGOA

OB30:



Nombre	Tipo de datos	Valor predet.	Comentario
1	Input		
2	Initial_Call	Bool	Initial call of this OB
3	Event_Count	Int	Events discarded
4	Temp		

```

//PID-AREN ALGORITMOA
2 "PID_DB" (Kc:=1.50,
3 I:=3,
4 Ti:=36.762,
5 Id:=0,
6 N:=20,
7 sentsorea_int:="Sentsore_int",
8 setpoint_I:="setpoint",
9 GOI_LIM_ST:=100.0,
10 BEHE_LIM_ST:=0.0,
11 GOI_LIM_SV:=5.0,
12 BEHE_LIM_SV:=0.0,
13 OP_V:=2.0,
14 AUTOMATIKO:="OP_BOOL",
15 u_analog:="Berogailu_INT",
16 sentsore_I:="Sentsore_I",
17 u:="u_V");
18
19 "setpoint" :=50.0;
20
21
  
```

FB1:

```

1 //SARRERAKO ALDAGAIEN KONBERTSIOA
2 //SENSOREA
3 #Karakterizazio_faktora := 10.0 / #GOI_LIM_SV;
4 #GOI_LIM_ST_karak := #Karakterizazio_faktora * #GOI_LIM_ST;
5 #error1:=SCALE(IN := #sentsore_int, HI_LIM := #GOI_LIM_ST_karak, LO_LIM :=#BEHE_LIM_ST, BIPOLAR := 0, OUT => #sentsore_I);
6 #sentsore_V := #sentsore_I * #GOI_LIM_SV / #GOI_LIM_ST;
7 //SETPOINT
8 #setpoint_V := #setpoint_I * #GOI_LIM_SV / #GOI_LIM_ST;
9 //OPERAZIO PUNTUA
10 IF #AUTOMATIKO=0 THEN
11   #u := #OP_V;
12 ELSE
13   //PID ALGORITMOA V UNITATETAN
14   #e := #setpoint_V - #sentsore_V;
15   #up := #Kc * #e;
16   #ui := #u1 + #Kc * (#I / (2 * #Ti)) * (#e + #e1);
17   #ud := (#ud1 * #Id + #Kc * #Id * #N * (#e - #e1)) / (#N * #I + #Td);
18   #u := #up + #ui + #ud;
19   //ALDAGAIEN AKTUALIZAZIOA
20   #u1 := #ui;
21   #ud1 := #ud;
22   #e1 := #e;
23
24 END_IF;
25
26 //IRTEERAKO ALDAGAIEN KONBERTSIOA IRTEERA ANALOGIKOA
27 #u_N := #u * 400.0 / 5.0;
28 #GOI_LIM_SV_karak := #Karakterizazio_faktora * #GOI_LIM_SV;
29 #error2 := UNSCALE(IN := #u, HI_LIM := #GOI_LIM_SV_karak, LO_LIM := #BEHE_LIM_SV, BIPOLAR := 0, OUT => #u_analog);
30
31
  
```