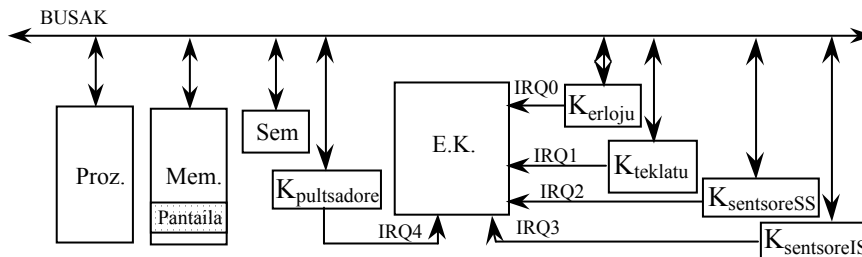


irteeraordua, zenbatekoa) errutina daukagu. Honek, zenbatekoa kalkulatzear gain, pantaila txiki batean aurkezten du eta *zenbatekoa* aldagaian itzultzen du. Dirua jaso ondoren, zaindaria 'L' tekla sakatuko du eta irteerako langa altxatu egin beharko da **altxatu_irteera_langa()** errutinaren bitartez. Oraingoan 5 segundo mantenduko da altxatuta, eta denbora-tarte horren ondoren jaitsi egin beharko da **jaitsi_irteera_langa()** errutinaren bitartez. Horrez gain, ibilgailu bat ateratzen den bakoitzean ordenadorearen pantailako azken lerroan ibilgailuaren datuak idatzi beharko ditugu: matrikula, sarrerako ordua, irteerako ordua eta zenbatekoa, horretarako **idatzi (matrikula,sarreraordua,irteeraordua,zenbatekoa)** errutina erabiliz informazio guztia eman behar zaiolarik. Informazioa idatzi ondoren soinu txiki bat sortuko da **beep()** errutinaren bitartez. Atera diren azken 25 ibilgailuen datuak beti pantailan izatea nahi dugu.

Idatzi lengoia algoritmikoan K_{IN} , K_{OUT} , K_{DMA} eta erlojuaren zerbitzu-errutinak hala nola sistemako programa nagusia. Aipatu eten-bektoreko zein osagaiak aldatu behar diren.

-----000000000000-----

2.-Aireportu bateko hegazkinen aieratze eta lurreratze gunean **hegaztiak** dauden kontrolatzen duen sistema daukagu. Sistemaren hardware egitura klasikoa da, 2. irudian ikus dezakezunez, ondorengo berezitasunak dituelarik: bi sentsore (SS eta IS) eta pultsadore bat. Osagai hauen funtzionamendua geroxeago azalduko da. Hauetaz gain, semaforo bat kontrolatu behar da; honen kontroladoreak kontrol-erregistro bat baino ez dauka memorian mapeatuta dagoena, SEM helbidean hain zuzen.



2. irudia: Hegaztien presentzia kontrolatzen duen sistemaren hardware egitura.

Mikroprozesadoreak kontrolatu behar dituen periferikoak klasean ikusitakoak dira, bi sentsoreak, pultsadorea eta semaforoa izan ezik. Hauen ezaugarriak ondoren aipatzen dira:

- * **SS sentsorea:** Hegaztien Sarrera Sentsorea. Kontrolpeko zonara hegaztiren bat edo gehiago sartzen den detektatzen du eta ondorioz eten-eskaera sortzen du. Bere datu-erregistroan une horretan sartu diren hegaztien kopurua irakur daiteke. Datu-erregistroa irakurri ondoren, *strobe* segida bat egin behar da bere kontrol-erregistroan.
- * **IS sentsorea:** Hegaztien Irteera Sentsorea. Kontrolpeko zonatik hegaztiren bat edo gehiago ateratzen den detektatzen du eta ondorioz eten-eskaera sortzen du. Bere datu-erregistroan une horretan atera diren hegaztien kopurua irakur daiteke. Datu-erregistroa irakurri ondoren, *strobe* segida bat egin behar da bere kontrol-erregistroan.
- * **SEM semaforoa:** Hegazkinei abisatzeko semaforoa. Berde egongo da kontrolpeko zonan hegaztirik ez dagoenean, beraz hegazkinak maniobratu dezake arriskurik ez dagoelako. Alderantzizko kasuan, **Gorri** egongo da. Lehentxeago aipatu denez, memorian mapeatuta dagoen kontrol-erregistroa baino ez dauka.
- * **Pultsadorea:** Eten-eskaera bat sortzen du hegaztiak ihes arazi behar dituen langileriak sakatzen duenean. Pultsadorea sakatuko dute hegazti bat harrapatzen duten bakoitzean.
- * **Erlojua:** 18 aldiz segundoko eteten du prozesadorea.
- * **Teklatua:** Tekla bat sakatzen denean, bi eten-eskaera sortzen ditu: lehenengoa tekla sakatu (MAKE) dela adierazteko eta bigarrena, askatu (BREAK) dela. Datu-erregistroan azkenekoz sakatutako tekla dagokion scan-kodea irakur daiteke eta kontrol-erregistroan *strobe* sekuentzia bat idatzi behar da eten bat onartzen denean. Ez dauka egoera-erregistrorik.
- * **Etenen kontroladorea:** Ondoko erregistro hauek ditu: IMR maskara-erregistroa, IRR eten-eskaeraren erregistroa eta ISR zerbitzuan dauden etenen erregistroa.

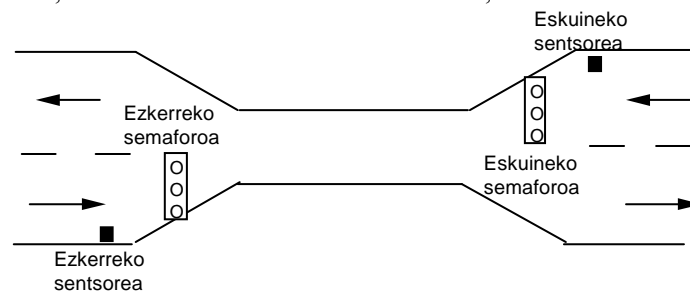
Sistemaren **funtzionamendua** hauxe da. **Egoera arrunta edo normala** kontrolpeko zonan hegaztirik ez dagoenean ematen da eta horrexegatik semaforoa **berde** egongo da. Beste edozein kasutan, semaforoa gorri egongo da. Hegaztien sarrera detektatzean, sistema **aurre-alarma** egoerara igaroko da; egoera honetan mantenduko da kontrolpeko zonan hegaztiak dauden bitartean (garbi dago hegaztiak sartu diren bezala atera daitezkeela, hots "de motu propio") eta sistemaz arduratzen den langileak hegaztiak uxatzeko premia adierazten ez duen bitartean. Premiazkoa izango da hegaztiak uxatzea hegazkin batek maniobraren bat egin behar duenean (nahiz lurreratu nahiz aineratu). Kasu honetan langileak **A tekla** sakatuko du; honen ondorioz sistema **alarma** egoerara igaroko da.

Alarma egoeran, semaforoa gorri mantentzeaz gain, soinu-alarma bat sortzen da (suposatu *sortu_alarma()* errutina jadanik idatzita daukagula); soinuak, hegaztiak uxatzeaz aparte, berauek harrapatu behar dituen langile-koadrilari abisatzen dio. "Garbiketa brigada" honek kontrolpeko zonatik alde egiten ez duten hegaztiak sare batez harrapatu behar ditu. Hegazti bana harrapatzean pultsadorea sakatu behar dute sistemari hegazti bat gutxiago dagoela adierazteko. Brigada honek kontrolpeko zona hegaztiz azkeneko garbitzen duenean, sistema **atze-alarma** egoerara igaroko da. Egoera honetan 2 minutu mantenduko da. Denbora-tarte honetan berriro hegaztiak sartzen badira, alarma egoerara itzultzen da. Bestela, denboraldi hori igarotzen bada hegaztirik gabe, egoera arruntera itzultzen da.

Idatzi lengoaia algoritmikoan ondoko zerbitzu-errutinak: SS sentsoarena, IS sentsoarena, pultsadorearena, teklatuarena, erlojuarena.

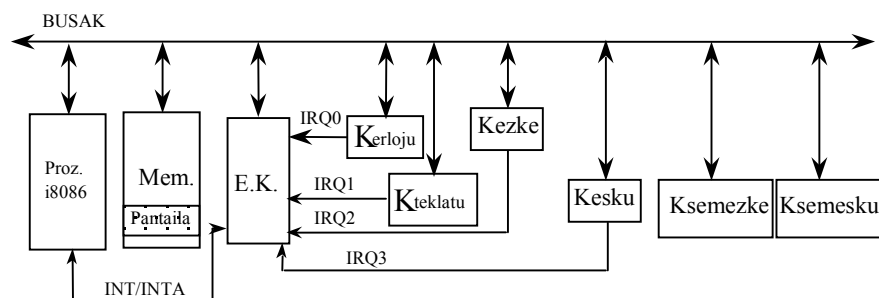
-----000000000000-----

3.-Herri txiki batean **zubi** arazotsu bat daukate: bere estatusuna dela eta autoak norabide bakar batean pasa daitezke. Errepedea bi norabidekoa denez, zubiaren muturretan semaforoak daude, eskematxoa ondokoa izanik:



3a. irudia: Zubiaren eskema.

Semaforo hauen kontrola automatikoki egin nahi dugu. Hardware egitura 3b. irudian ageri da.



3b. irudia : Sistemaren hardware egitura.

Mikroprozesadoreak kontrolatu behar dituen periferikoak, klasean ikusitakoak dira, ondokoak izan ezik:

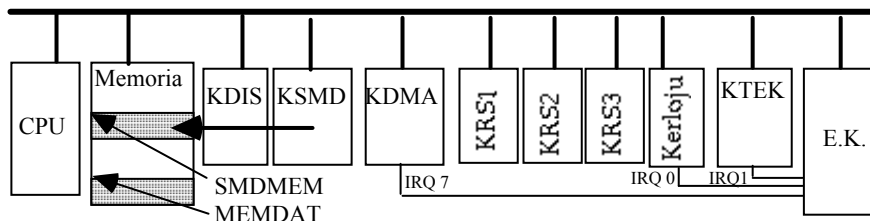
- * **Kezke** eta **Kesku** Bi aldeetan dauden sentsoaren kontroladoreak dira. Bere parean auto bat geldirik baldin badago semaforoaren zain, eten-eskaera bat sortzen dute 2 segundoro. 3b. irudian ikus daitezke zer IRQ lerrotik eteten duen bakoitzak.
- * **Ksemezke** eta **Ksesesku** Bi aldeetan dauden semaforoaren kontroladoreak dira. **Datu-erregistro** eta **kontrol-erregistro** bana dute (**RDAT_EZKE**, **RKON_EZKE**, **RDAT_ESKU** eta **RKON_ESKU**). Datu-erregistroan, semaforoaren kolorea adierazten da (gorria, laranja edo berdea) eta kolore hori aldatzerakoan STROBE sekuentzia egin behar da kontrol-erregistroaren 4 bitean.

Sistemaren funtzionamendua ondokoa izango da. Sistema hasieratzerakoan bi semaforoak gorrian jarriko dira, eta ezingo dira egon inoiz biak batera berdean. Ezkerraldean (edo eskuinaldean auto bat (edo gehiago) detektatzen bada alde horretako semaforoa berdean jarriko dugu, kolore horretan mantenduz 10 segundo. Denbora-tarte horretan eskuinaldean (edo ezkerraldean) autoak geldirik badaude ezin da alde horretako semaforoa berdean jarri, aipatu baitugu inoiz ezin direla bi semaforoak berdeak egotea une berean. 10 segundo horiek igaro ondoren, ezkerraldeko semaforoa (edo eskuinekoa) laranja jarri behar da 2 segundoz. Semaforo bat laranja dagoen bitartean beste aldeko eskakizunei emango diegu lehentasuna. Beraz, 2 segundo horien zehar, eskuinean autoak detektatzen badira (edo ezkerrean), zera egin behar da, 1,5 segundo itxaron eta ondoren eskuineko (edo ezkerreko) semaforoa berdean jarri eta ezkerrekoa berriz (edo eskuinekoa) gorrian. Aipatu ditugun 2 segundo horietan, kontrako aldean autorik ez badago, ezkerreko (edo eskuineko) semaforoa berriz gorrian jarriko dugu, horrela biak gorriak izanik berriz. Bestalde, gure kontrol-programa amaitu egin behar du, 'B' tekla sakatu dela detektatzen bada. Teklatuaren sinkronizazioa inkestaz egingo da programa nagusian.

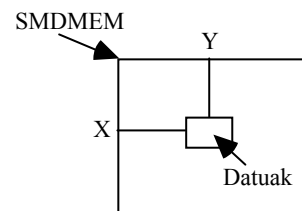
Idatzi lengoia algoritmikoa: a) K_{ezke} , K_{esku} eta erlojuaren zerbitzu errutinak eta b) programa nagusia.

-----0000000000-----

4.-Informatika Fakultateko Arkitekturako ikasleek eskaera berri bat jaso dute Granada-Dakkar Rallyaren antolatzaileengandik. Hauek “**satelite bidezko posizionamendu globala**” izeneko sistema (GPS) garatu nahi dute, gidariak gal ez daitezzen basamortuan. GPS sistemaren bitartez, lor daitezke uneko puntu baten koordenatuak, latitudea (X) eta longitudea (Y), hau guztia satelite-sistema batean oinarrituz. Horrez gain, memorian mapa digital bat baldin badu, puntu horri buruzko informazio gehiago eman dezake (altuera, landaretza, bideak, ...). Sistemak ondoko sarrera/irteerako periferikoak ditu:



SMD: Mapa digitalaren irakurtzaile/scanner-a. Honen bitartez mapa batek dituen datuak sar daitezke. Bere kontroladoreak, K_{smd} , maparen datuak 256×256 osagaiko bektore batean uzten ditu, memorian mapeatuta SMDMEM posiziotik aurrera. Bektorearen osagai bakoitzak puntu bati buruzko 4 byteko informazioa gordetzen du (altuera, luraren ezaugarriak, ...).



Sateliteen hiru hartzaile $RS(i)$, non $i: [1..3]$: Hauetako bakoitzak satelite baten seinaleak jasotzen ditu. Bere kontroladoreek, $KRS(i)$, ondoko erregistroak dituzte:

- RDatSat(i):** Sateliteak bidalitako posizio-kodea gordetzen duen datu-erregistroa da.
- REgoSat(i):** Egoera-erregistroa da eta datua prest dagoela jakiteko lekoa eduki arte itxaron behar da
- RKonSat(i):** Kontrol-erregistroa da eta datu bat irakurtzen denean abisatu behar da bertan lekoa bat idatziz (*strobe*).

Teklatua TEK: Klasean ikusi dugun bezala funtzionatzen du. Kasu honetan, hiru tekla besterik ez dira erabiliko, geroago ikusiko dugun bezala.

“Display”-aren kontroladorea, Kdis: Kristal likidozko display txiki batean puntuaren X eta Y koordenatuak, berari buruzko informazioa eta mezu bat idatz daitezke. Horretarako ondoko erregistroak ditugu:

- DispX:** X koordenatua
- DispY:** Y koordenatua
- DispDat:** X, Y puntuari lotutako datuak
- DispMez:** Mezua
- DispKon:** Kontrol-erregistroa. Datua prest dagoenean lekoa idatzi behar da bertan.

Memoriako atzipen zuzenekoaren kontroladorea, **Kdma**: Bere erregistroak ondokoak dira:

- @JatDMA:** transferitu behar den datuaren helbidea gordetzen du
- @HelDMA:** helburuko helbidea gordetzen du
- LuzDat:** transferitu behar den datu-kopurua (bytetan)
- KonDMA:** DMA transferentzia hasteko 1era jarri behar da
- EgoDMA:** Iekoa baldin badu transferentzia ondo joan da.

Funtzionamendua. Teklatua etenen bitartez sinkronizatzen da eta hiru tekla besterik ez ditu: **A** = Posizioa, **B** = Mapa berria irakurri eta **C** = Bukatu.

A: Posizioa sakatzen denean, **INKESTA BITARTEZ** hiru sateliteen kodeak irakurtzen dira, eta **datsat(i)** aldagaien uzten dira. Inkesta bakoitzak ez du 30 segundo baino gehiago iraun behar. Kodeak ditugunean, **posizioa-kalkulatu** (*datsat, Px, Py*) errutinak, sateliteen datuak erabiliz, puntuaren koordenatuak itzultzen ditu Px eta Py parametroetan.

Hauek ditugunean, pantailan idatzi behar da:

- Px eta Py koordenatuak
- memoriako mapak duen puntu horri buruzko informazioa
- mezu bat, bere edukia ondokoren bat izango delarik:
 - "ZUZENKI", hiru sateliteen datuak baldin badaude.
 - "GUTXI GORA BEHERA", bi sateliteekin bakarrik komunikatu badugu.
 - "KOMUNIKAZIOARIK EZ" bi satelite baino gutxiago igortzen dutenean. Kasu honetan Px=0 eta Py=0 izango dira.

B: Mapa irakurri sakatzen denean informazioa transferitu behar da scanner-aren kontroladoretik, Ksmc, memoriako bektore batera. Honen tamaina 256*256 da eta posizioa-kalkulatu errutinak lan egiten duen espazioan kokatzen da, **MEMDAT** helbidetik aurrera hain zuzen (hau da *posizio-kalkulatu* errutinaren laneko helbidea).

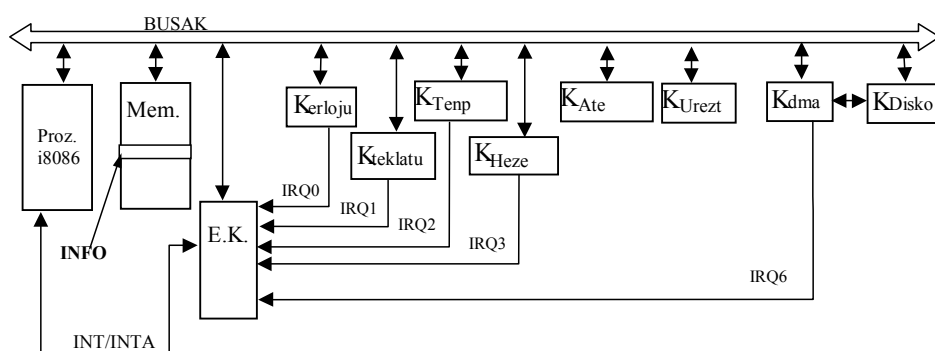
C: Bukatu, programak amaitu egin behar du.

OHAR GARRANTZITSUA: tekla bati ez zaio kasurik egingo aurretik sakatutako teklari dagozkion ekintza guztiak bete arte.

Eskatutakoa: a) programa nagusia eta b) ondorengo zerbitzu-errutinak: erlojua, teklatua eta DMA.

-----000000000000-----

5.- Negutegi bateko tenperatura eta hezetasuna automatikoki kontrolatzen dituen sistema bat diseinatu nahi dugu. Sistema honek bi sentsore ditu, bat tenperaturaren kontrolerako eta bestea hezetasunaren kontrolerako, hurrengo irudian ikus daitekeen bezala.



- K_TENP:** Tenperatura-sentsorearen kontroladorea da. Tenperatura aldatzen den bakoitzean eteten du eta tenperaturaren balio berria datu-erregistroan gordetzen du (RDAT_TENP). Datu-erregistroa irakurri ondoren STROBE sekuentzia egin behar da kontrol-erregistroan (RKON_TENP).
- K_HEZE:** Hezetasun-sentsorearen kontroladorea da. Hezetasun erlatiboa balio minimo bat (HEZ_MIN) baino baxuagoa denean eteten du.
- K_ATE:** Negutegiaren ateen kontroladorea da. Kontrol-erregistro bat du (RKON_ATE) bertan adierazteko burutu nahi den ekintza: Iekoa idatziz atea irekitzen dira eta 0koa idatziz itxi egiten dira. Oharra: suposatuko dugu ez dela ezer gertatzen atea irekitzen den egonik berriro irekitzen bagara edo irekita egonik berriro irekitzen saiatzen bagara.

- K_UREZT:** Ureztaketarako iturrien kontroladorea da. Kontrol-erregistro bat du (RKON_UREZT) burutu nahi den ekintza adierazteko: lekua idatziz iturriak irekiko dira eta 0koa idatziz itxi egingo dira. Oharra: suposatuko dugu ez dela ezer gertatzen iturria itxita egonik berriro ixten saiatzen bagara edo irekita egonik berriro irekitzen saiatzen bagara.
- K_DMA:** DMA kontroladorea da. Memoria eta kanpoko disko-unitate baten artean transferentziak burutzeko erabiltzen da. Ondorengo erregistroak ditu:
- **Kontrol-erregistroa** (RKON_KDMA): erregistro honetan lekua idaztean transferentzia hasiko da.
 - **Helbide-erregistroa** (RHEL_KDMA): bertan adierazi behar da memoriako zein helbidetik aurrera metatzen den diskora transferituko den blokea.
 - **Luzera-erregistroa** (RLUZ_KDMA): bertan adierazi behar da transferituko den blokearen luzera bytetan.
 - **Egoera-erregistroa** (REGO_KDMA): transferentzia amaitzerakoan, erregistro honen edukiak transferentzia ondo joan den (1) edo errorenen bat gertatu den (0) adierazten du.
- K_DISKO:** Diskoaren kontroladorea da. Transferentzia bete ahal izateko kontroladore hau hasieratu behar da. Suposatuko dugu *KdiscoProgramatu()* izeneko errutina daukagula hasieratze hori egiteko.

Sistemaren **funtzionamendua** ondorengoa izan beharko da. Tenperaturak balio maximo bat gainditzen duenean (TENP_ALTUA), negutegiaren atea ireki behar dira. Ateak berriro itxi behar dira tenperatura egokitzat onartzen den balio bat baino txikiagoa denean (TENP_ITXI). Bestalde, tenperatura balio minimo bat baino txikiago dela detektatzen bada (TENP_OSΟ_BAXUA), alarma bat sortu behar da (*SortuAlarma()* errutinaren bitartez) negutegiko langileei abisatzeko. Hortik aurrera sistema “geldiarazi” beharko da eta ez da beste kontrolik egingo (ez tenperatura eta ez hezetasun kontrolik) langileak arazoa konpondu arte. Dena konpontzerakoan, langileak ‘C’ tekla sakatuko du, eta sistema hasierako egoera batetara itzuliko da atea itxiz. Hezetasun erlatiboa minimoa baino baxuagoa dela detektatzen bada, ureztaketa iturriak ireki behar dira, 5 minututan zehar irekita mantenduz. 5 minutu hauetan ez da egingo ez tenperaturaren ez hezetasunaren kontrolik.

Tenperatura eta hezetasunari buruz lortzen den informazio guztia gorde nahi da historiko gisa. Horretarako *GertaeraGorde(Gertaera_zkia, Gertaera_mota, Balioa, Ordua)* funtzioa erabiliko dugu. *Gertaera_zkia* zenbatgarren gertaera den adierazten duen zenbakia da; *Gertaera_mota* parametroak bi balio izan ditzake TENP edo HEZET; *Balioa* tenperaturaren balioa edo 0 (HEZET gertaera-motaren kasuan) izan daiteke; *Ordua* gertaera detektatu deneko ordua da. Informazio hori bektore batean gordetzen da, honen hasierako helbidea INFO izanik. Gertaera bakoitzerako gordetzen den informazioa 40 bytekoa da.

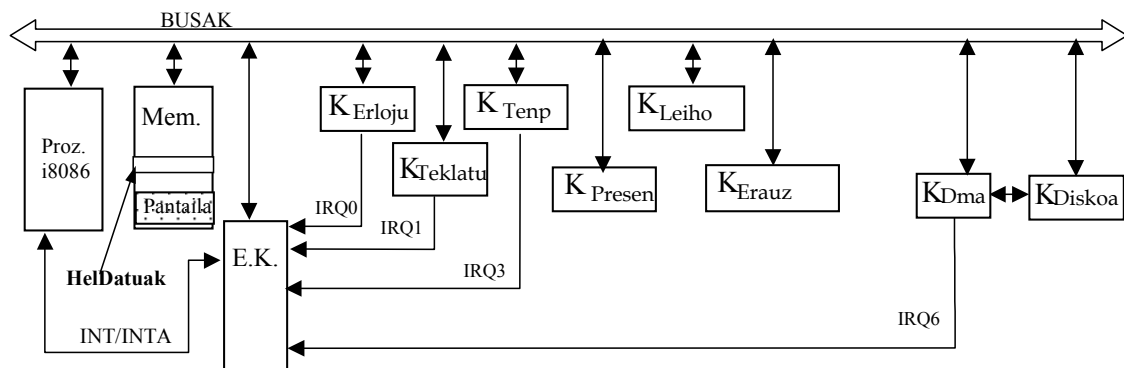
Edozein unetan langileak aplikazioaren bukaera eska dezake ‘S’ tekla sakatuz. Kasu honetan, bukatu aurretik, gertaeren informazioa diskora transferitu behar da. Ez dira 3 saio baino gehiago egingo transferentzia ongi ez bada. Hirugarren errorea suertatzen bada, errore-mezu bat emango da programatuta dagoen *ErroreaTransferentzian()* errutina erabiliz eta aplikazioa bukatuko da.

Ordua, ORDUA aldagai orokorrean aurkitzen da. Bere tratamendurako programatuak dauden ondorengo funtzioak erabil ditzakezu: *OrduaHasieratu(ORDUA)* sistemaren hasieran ordua hasieratzeko eta *OrduaEguneratu(ORDUA)* ordua segundoro eguneratzeko. Gailu guztien erregistroak S/I espazioan daude (ez daude memorian mapeatuta). Erlojuaren, teklatuaren eta etenen kontroladoreak klasean ikusitakoak dira. **Teklatuaren sinkronizazioa inkesta** bidez egingo da.

Hurrengoa eskatzen da: Idatzi lengoaia algoritmikoan beharrezkoak iruditzen zaizkizun zerbitzu-errutina guztiak eta programa nagusia. Komentatu ariketaren ebazpenerako egiten duzun edozein suposaketa.

-----000000000000-----

6.- Fakultateko eta gelategiko zenbait gela eta bulegoren tenperatura altua dela eta, errektoreordetzak **tenperatura kontrolatzeko sistema automatiko** bat erabiltzearen abantaila aztertzea eskatu digu, geletako batean sistema probatuz. Horretarako gelan zenbait gailu berri jarri dira: *tenperatura-sentsoreak* gelan dagoen tenperatura ezagutzeko, *presentzia-sentsoreak* gelan norbait dagoen edo ez jakiteko, *motoreak* leihoak irekitzeko edo ixteko eta gela aireztatzeko *erauzgailuak* (“*extraktoreak*”). Ondorengo irudian sistemaren egitura ageri da:



KTemp: Gelako temperatura-sentsorea da. Tenperaturaren balioa zein den adierazteko aldizka etengo du. Etenen duenean, RDAT_KTemp datu-erregistroan tenperaturaren balioa ematen du. Erregistroaren balioa irakurri ondoren *strobe* sekuentzia bat egin behar da kontrol-erregistroan (RKON_KTemp).

KPresen: Gelako presentzia-sentsorea da. Datu-erregistroaren bidez gelan norbait dagoen edo ez adierazten digu (1 sala okupatua, 0 sala hutsa). Gailu honen **sinkronizazioa inkesta bidez** burutu behar da.

KLeiho: Gelako leihoen motoreak gailu honetan zentralizatu dira. Leihoak irekitzeko edo ixteko datu-erregistroan (RDAT_KLeihio) eragiketa adierazi behar da (1 ireki, 0 itxi) eta kontrol-erregistroan (RKON_KLeihio) *strobe* sekuentzia egin behar da.

KErauz: KLeihio kontroladorerako deskribatutako funtzionamendu bera du. Kasu honetan datu-erregistroa RDAT_KErauz da (1 aktibatu, 0 desaktibatu) eta kontrol-erregistroa RKON_KErauz.

KDMA: DMA kontroladorea da eta memoria eta disko-unitate baten artean transferentziak egiteko erabiltzen da. Ondorengo erregistroak ditu:

Kontrol-erregistroa (RKON_KDMA). Erregistro honetan leku bat idaztean transferentzia hasten da (automatikoki jartzen da 0ra transferentzia bukatzen denean).

Helbide-erregistroa (RHEL_KDMA). Transferitu behar den blokearen memoriako hasiera-helbidea gordetzen du.

Luzera-erregistroa (RLUZ_KDMA). Transferitu behar den blokearen luzera gordetzen du.

Egoera-erregistroa (REGO_KDMA). Behin transferentzia bukatu denean, erregistro honen edukiak transferentzia ondo joan den (1) edo erroren bat gertatu den (0) adierazten du.

KDisko: Diskoaren kontroladorea da. Kontroladore hau programatzeko transferentzia bat egin baino lehen, **Kdisko_programatu()** errutina daukagu.

Sistemaren funtzionamendua ondorengoa da. Gelan norbait dagoenean, sistema gelako temperatura egonkortzen eta balio normal batean mantentzen saiatu behar da. Gela hutsik badago, edo momenturen batean husten bada, ez da ezeren kontrolik egingo. Gelan norbait dagoela detektatzean, sistemak tenperaturaren kontrolari ekingo dio. Gelako tenperaturak TENP_ALTUA temperatura jakin bat gainditzen bada, leihoak ireki behar dira temperatura jaisteko. 5 minutu pasa ondoren gelako temperatura ez bada jaitsi TENP_ALTUA baliotik, erauzgailuak aktibatu behar dira gela aireztatzeko. Hala ere, beste 5 minutu pasa ondoren, temperatura TENP_ALTUA gainetik badago, alarma bat aktibatu behar da programatua dagoen *Alarma ("Temperatura altuegia")* funtzioa erabiliz. Une horretan, gainbegiratzailerak gela hustu eta 'A' tekla sakatuko du. Tekla hau sakatzean, *AlarmaDesaktibatu()* funtzioa erabiliz alarma desaktibatu, leihoak itxi eta erauzgailua desaktibatu behar dira. Alarma sortu aurretik temperatura TENP_ALTUA baliotik jaisten bada, leihoak itxi edota erauzgailuak desaktibatu behar dira. Momenturen batean gela hutsik geratzen bada, kontrolak egiteari utzi behar zaio, leihoak itxiz edota erauzgailuak desaktibatuz.

Minuturo gelako temperatura memorian metatu behar da *TemperaturaGorde(Temperatura, Ordua)* funtzioa erabiliz, non temperatura azkeneko aldiz detektatutako temperatura den eta ordua ORDUA aldagaian dagoen. Gauero 00:00-ak iristean, *TemperaturaGorde* funtzioa erabiliz gorde den informazio guztia diskora pasa behar da. Informazioa memorian gordeta dago *HelDatuak* helbidetik aurrera eta erregistroak neurri finkokoak dira (100 byte). Transferentzia honetan erroren bat gertatzen bada mezu bat atera behar da pantailatik *ErroreMezua("DMA errorea")* funtzioa erabiliz, baina sistemak normal funtzionatzen jarraituko du.

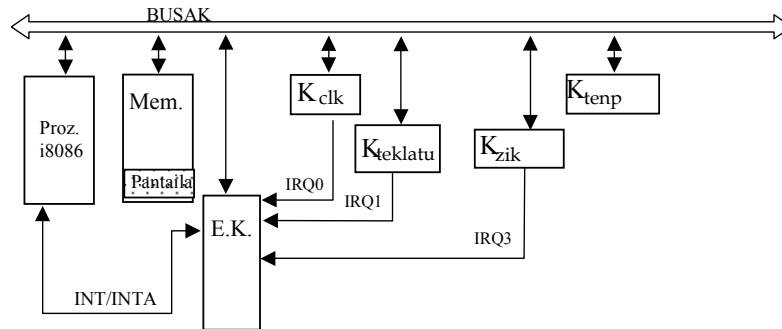
Ordua, ORDUA aldagai orokorrean aurkitzen da. Bere tratamendurako programatuak dauden ondorengo funtzioak erabil ditzakezu: *OrduaHasieratu(ORDUA)* sistema abiatzen denean ordua hasieratzeko eta

OrduaEguneratu(ORDUA) ordua segundoro eguneratzeko. Erlojuaren eta teklatuaren kontroladoreak klasean ikusitakoak dira. Kasu honetan teklatuaren sinkronizazioa etenen bidez burutzea nahi da. Gailu guztien erregistroak S/I-n mapeatuta daude.

Ondorengo eskatzen da: Idatzi lengoia algoritmikoan beharrezkoak iruditzen zaizkizun zerbitzu-errutina guztiak eta programa nagusia. Komentatu ebazpenean egiten duzun edozein suposaketa.

-----0000000000-----

7.- Itsasoaren mailan dagoen arrain-haztegi baten ura kudeatzeko sistema diseinatu nahi da. Horretarako, hurrengo irudian adierazten den hardware egitura kontuan hartuko da:



Periferiko berrien kontroladoreak hurrengoak dira:

- Kzik:** Arrain-haztegiaren azalean zikinkeria maila altua dela detektatzen duen kontroladorea da. Zikinkeria mailak aurrez jarritako neurri bat gainditzen duenean eteten du. Eteten duen guztietan STROBE bat egin behar da bere kontrol-erregistroaren (RKON_KZIK) pisu gehieneko bitean.
- Ktenp:** Uraren tenperatura neurtzen duen kontroladorea da. REGO_KTENP egoera-erregistroan 1 balioa dagoen bakoitzean, RDAT_KTENP datu-erregistroan urak dituen Celsius graduak prest daudela zehazten da. RKON_KTENP kontrol-erregistroaren pisu gehieneko bitean STROBE bat egin behar da datu-erregistroa irakurtzen den bakoitzean.

Erlojuaren, teklatuaren eta etenen kontroladoreak klasean ikusitakoak dira eta sistema honetan **teklatuaren sinkronizazioa etenen bidez** burutuko da. Uraren **tenperaturaren analisia** berriz, **inkestaz** egingo da. Teklatuaren kontroladoreari dagokion erregistroak RDAT_KTEK eta RKON_KTEK dira, hurrenez hurren, datu eta kontrol-erregistroentzat.

Sistemaren funtzionamendua honakoa da. Igerilekuan tenperaturari dagokionez arazorik ez badago, bost minuturo uraren tenperatura aztertu behar da. Ura -5° Celsius eta $+15^{\circ}$ Celsius artean baldin badago, arrainak normaltasun osoz bizi direla suposatzen da eta ez da aparteko neurririk hartu behar. Baina tenperatura -5° tik jaisten bada, ura berotu behar da, bestela arrainak hil baitaitezke denbora-tarte bat pasa eta gero. Ura berotzeko, honakoa egiten da: *ur_beroa()* errutinaren bidez 100 litro ur bero botatzen dira igerilekura. 100 litroak erantsi ondoren, tenperaturak hotza jarraitzen badu, behin eta berriz ur beroa eransten jarraitu behar da *ur_beroa()* errutina erabiliz. Ura $+15^{\circ}$ tik gora egoteak ere arrainak hil ditzake. Horren aurrean, ura hoztuko da *izotzak()* errutinaren bidez 100 litro izotz erantsiz. Hemen ere, 100 litroak erantsi eta gero, tenperaturak bero jarraitzen badu, izotzak erantsiko dira behin eta berriz tartera itzuli arte. Ikus daitekeenez, arazoren bat dagoen bitartean, tenperatura behin baino gehiagotan aztertzen egon behar da. Arazoa konpondu eta gero, hortik bost minutura aztertuko da berriz tenperatura.

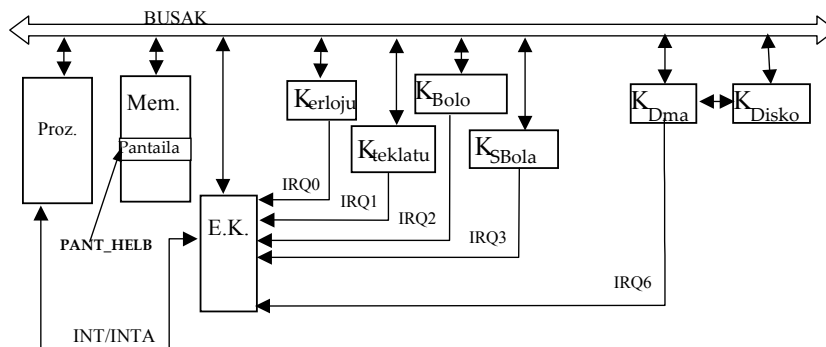
Tenperaturaz gain, ura zikin egoteak arrain asko hiltzea ekar lezake. Aipatu den bezala, zikinkeriaren kontroladoreak etena eskatzen du azaleko zikinkeriak neurri bat gainditzen duenean. Kasu horretan, alarma akustiko bat martxan jartzen da (*alarma()* errutina) eta horregatik haztegiko langileak zikinkeria batzen hasten dira igerilekuaren azala zeharkatzen duten sare batzuen bidez (sareak martxan jartzen dira *sareak_aktibatu()* errutinaren bidez). Zikinkeria langileek igerilekuaren alboan dauden kontainer batzuetara botatzen dute (suposatu kontainerren kopurua eta beraien edukiera arazoari aurre egiteko nahikoak direla). Langileen nagusiak igerilekuaren azala modu egokian garbitu dela erabakitzen duenean, sistema kontrolatzen duen PCan 'D' sakatuko du, eta horren ondorioz alarma isiltzen da (*alarma_gelditu()* erabiliz), sareak gelditzen dira (*sareak_gelditu()* dela medio) eta zikinkeria gordetzen duten kontainerrak beste batzuegatik aldatzen dira, *kontainer_aldatu()* erabiliz. Edozein kasutan, kutsadura akustikoa saihesteko, alarma gehienez 30 segundoz jotzen egongo da behin martxan jarri ondoren. Tarte hori gainditzen bada, alarma gelditu egin behar da.

Ondorengo eskatzen da:

- (a) Idatz itzazu lengoaia algoritmikoa beharrezkoak iruditzen zaizkizun zerbitzu-errutina guztiak eta programa nagusia. Komentatu ariketa ebazteko egiten duzun edozein suposaketa.
- (b) Azaldu argi eta garbi teklatuaren sinkronizazioa inkesta bidez egiteak suposatuko lituzkeen aldaketak.

-----0000000000-----

8. Bolatoki batetako partidak kontrolatzen dituen sistema bat diseinatzea nahi dugu. Horretarako sentzore bat daukagu bola noiz jaurti den kontrolatzeko, eta gailu berezi bat erori diren boloak kontatu eta zutitzeko.



K_BOLO: Erori diren boloak zutitzeko erabiltzen den gailuaren kontroladorea da. Bi erregistro ditu: kontrol-erregistroa (RKON_KBOLO) eta datu-erregistroa (RDAT_KBOLO). Gailu hau martxan jartzeko strobe sekuentzia bat egin behar da bere kontrol-erregistroan. Strobe sekuentzia hori egitean gailuak botatako boloak altxatuko ditu. Boloak altxatu ondoren eten bat sortuko du eta bere datu erregistroan adieraziko du botatako bolo kopurua zein izan den.

K_SBOLA: Bola jaurtia izan dela detektatzen duen sentzorearen kontroladorea da. Bola boloetara iristear dagoen unean sentzore honen ondolik pasatzen da, eta kontroladoreak eten bat sortuko du.

K_DMA: DMA kontroladorea da. Memoria eta kanpoko disko-unitate baten artean transferentziak burutzeko erabiltzen da. Ondorengo erregistroak ditu:

- **Kontrol-erregistroa** (Rkon_kdma): erregistro honetan lekua idaztean transferentzia hasiko da. (Automatikoki jartzen da 0ra)
- **Helbide-erregistroa** (Rhel_kdma): bertan adierazi behar da memoriako zein helbidetik aurrera dagoen metatua diskora transferituko den blokea.
- **Luzera-erregistroa** (Rluz_kdma): bertan adierazi behar da transferituko den blokearen luzera bytetan.
- **Egoera-erregistroa** (Rego_kdma): transferentzia amaitzerakoan, erregistro honen edukiak adierazten du transferentzia ondo joan den (1) edo erroreren bat gertatu den (0).

K_DSKO: Diskoaren kontroladorea da. Transferentzia bete ahal izateko kontroladore hau hasieratu behar da. Suposatuko dugu *KdiskoProgramatu()* izeneko errutina daukagula hasieratze hori egiteko.

Sistemaren **funzionamendua** ondorengo izan beharko da. Hasiera batean, bolatokiko jabeak hasteko baimena eman arte ezingo gara jolasten hasi. Horregatik langa bat egongo da boloen aurrean. Behin jabeari ordaindu diogunean, honek teklatuko H tekla sakatuko du. Honela, langa altxatuko da eta partida has daiteke. Langa altxatzeko *altxatu_langa()* errutina jadanik prest dago.

Partida bakoitzak 10 jaurtiketa izango ditu eta jaurtiketa bakoitzean botatako boloak kontatuko dira. Jaurtiketa bakoitzean, SBOLA sentzoreak bola boloetara iristen ari dela detektatu eta eten bat sortuko du. Une horretatik 10 segundo pasa ondoren (eroritako boloak lurrean egonkortzeko denbora), BOLO gailuak boloak jaso beharko ditu. Jaurtiketa bakoitzaren ondoren, ordura arte guztira botatako bolo kopurua pantailaratu beharko da. Horretarako *pantailaratu_botatakoak(Bolo_Kopurua, Jaurtiketa_Zenbakia)* errutina erabiliko dugu, zeinak bi parametro jasotzen dituen: orain arte guztira bota den bolo kopurua, eta orain arteko jaurtiketa kopurua. Horrela, banan bana puntuazio guztiak bata bestearen ondoan pantailaratu joango dira. Puntuazio bakoitzak 4 byte okupatuko du memorian.

10 jaurtiketak burututakoan partida amaituko da, baina aurretik puntuazio guztiak gordeko dira diskoan. Memoriatik diskorako transferentzia honetarako DMA erabiliko dugu. DMA transferentzia ez bada ondo burutzen berriro saiaturiko

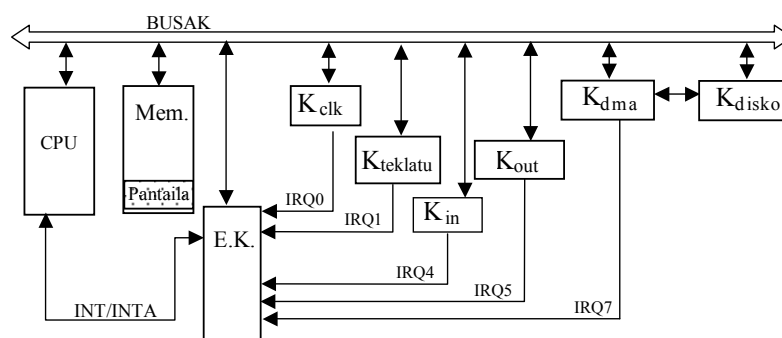
gara, baina 3 aldiz gehienez. 3 saiakeren ondoren transferentzia burutzea lortu ez badugu *errorea_transferentzian()* errutinari dei egingo diogu, eta partidari bukaera emango zaio. Partidak gehiegi ez luzatzeko asmoarekin, 5 minutuko denbora tartea dauka jokalariaik jaurtiketa bakoitza egiteko. Boloak altxatzen direnetik 5 minutu horiek pasatzen badira jaurtiketarik egin gabe, partida amaituko da. Azkeneko kasu honetan ez da puntuazioa diskora transferitu behar. Edozein kasutan partida amaitzean langa jaitsi beharko da berriro, teklatuan H tekla sakatu arte. Horretarako *jaitsi_langa()* errutina erabil dezakegu.

Pantaila memorian mapeatuta dago PANT_HELB helbidetik aurrera. Erlojuaren, teklatuaren eta etenen kontroladoreak klasean ikusitakoak dira. **Teklatuaren sinkronizazioa inkesta** bidez egingo da.

Hurrengoa eskatzen da: idatzi lengoaia algoritmikoan beharrezkoak iruditzen zaizkizun zerbitzu-errutina guztiak eta programa nagusia. Komentatu ariketaren ebazpenerako egiten duzun edozein suposaketa.

-----00000000000-----

9.-Parke tematiko batean, **beldurrezko korridore** bat erantsi behar dute. Korridore horretan pertsonak banan-banan sartu eta aterako dira. Sistemaren hardware egitura modu honetan geratzen da:



Periferiko berrien kontroladoreak hauek dira:

- Kin:** Sarrerako hesiaren kontroladorea da. Pertsona batek korridorerara sartu nahi duenean, hesiaren ondoan dagoen botoi bat sakatu behar du. Botoia sakatzean, eten bat sortzen du kontroladore honek. Bere datu-erregistroan (RDAT_Kin) sartu nahi duen pertsonaren pisua adierazten da eta bere kontrol-erregistroan (RKON_Kin) STROBEa egin behar da bere eten bat tratatzen den bakoitzean.
- Kout:** Irteteko atearen kontroladorea da. Korridorearen barruan dagoen pertsonak irteerako atea bultzatzen duenean, kontroladore honek eten-eskaera luzatzen du. Bere kontrol-erregistroan (RKON_Kout) STROBEa egin behar da bere eten bat tratatzen denean.
- Kdisko:** Diskoaren kontroladorea da. Transferentzia bete ahal izateko kontroladore hau hasieratu behar da. Suposatu *KdiskoProgramatu(eragiketa_mota)* izeneko errutina arduratzen dela hasieratze horretaz, egin nahi den eragiketaren arabera (diskotik IRAKURKETA ala diskoan IDAZKETA).
- Kdma:** DMA kontroladorea da. Memoriaren eta disko-unitatearen artean transferentziak egiteko erabiltzen da. Honako erregistro hauek ditu:
 - Kontrol-erregistroa (RKON_Kdma):** erregistro honetan lekoa idaztean transferentzia hasiko da. Automatikoki jartzen da 0an.
 - Helbide-erregistroa (RHEL_Kdma):** transferituko den blokearen hasierako helbidea.
 - Luzera-erregistroa (RLUZ_Kdma):** transferituko den blokearen luzera bytetan.
 - Egoera-erregistroa (REGO_Kdma):** transferentzia amaitzean, erregistro honek adierazten du transferentzia ondo joan den (0) edo errorenen bat gertatu den (1).

Erlojuaren, teklatuaren eta etenen kontroladoreak klasean ikusitakoak dira. Sistema honetan **teklatura inkesta bidez** tratatzen da. Teklatuaren kontroladorearen erregistroak RDAT_Kteklatu eta RKON_Kteklatu dira, datu-erregistroa eta kontrol-erregistroa, hurrenez hurren.

Sistemaren **funtzionamendua** honako hau izan beharko da. Pertsona batek korridorean sartu nahi badu, sarrerako hesiaren ondoan dagoen botoia sakatuko du. Korridorea ez badago hutsik, hesia ez da irekiko eta pertsonak botoia behin eta berriro sakatu beharko du hesia ireki arte. Korridorea hutsik badago, ordea, *sarrera_utzi()* errutinaren bidez hesia ireki eta pertsonari sartzen uzten zaio; bide batez, geroago erabiliko diren estatistika batzuetarako, pertsona horren pisua eskuratzen da.

Behin korridorearen barruan egonda, 5 segundoan behin, susto desberdinak emango zaizkio *susto()* errutinaren bidez, adibidez, barre-algara maltzurak entzungo dira, sugarrak botako dira, pizti zitalak agertaraziko dira, eta abar. *susto()* errutinak sistemak gordeta dituen sustoen artean ausazko aukeraketa egingo du exekutatzen den bakoitzean.

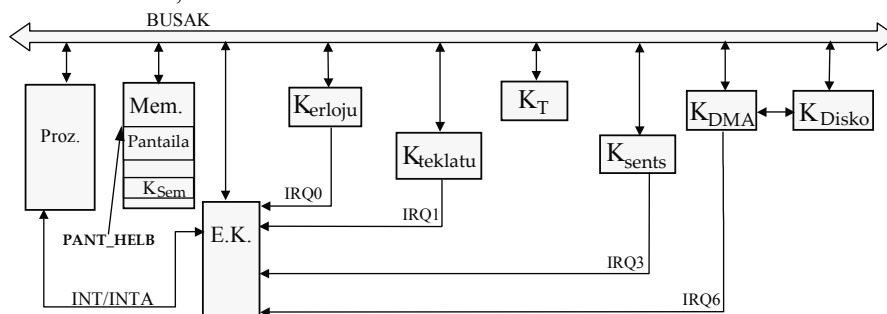
Pertsonak gehienez 3 minutuko tarte dauka korridorea zeharkatu eta amaierako atetik ateratzeko. 3 minutu pasatzen badira eta atetik ateratzea lortu ez badu, lurrean dagoen tranpa bat irekiko da, *tranpa()* errutinaren bidez, eta pertsona hori korridoretik kanporatua izango da, txirrista batetik eroriz. Baina 3 minutu baino lehen amaierako atetik ateratzen bada, memorian 40 byteko datu bat gorde behar da honako informazio honekin: zenbat pertsona atera diren amaierako atetik une horretara arte, atetik atera berri den pertsonaren pisua, eta korridorea zeharkatzeko behar izan duen denbora segundotan. Horretarako, *datuak_gorde(helbidea, pertsona, pisua, iraupena)* errutina erabiltzen da. Sistema hasieratu ondoren, amaierako atetik ateratzen den lehenengo pertsonari buruzko informazioa *Hashel* helbidean idazten da, eta hortik aurrera idatzi behar da amaierako atetik ateratzen diren hurrengo pertsonari buruzko informazioa, ondoz ondoko posizioetan.

Sistema gainbegiratzen ari den langileak 'D' tekla sakatzen duenean, korridorea hutsik badago, memorian gorde diren datuak diskora kopiatu behar dira. Baina 'D' sakatu duenean norbait baldin bazegoen korridorean, kopia hori egin behar da pertsona korridoretik ateratzean, atetik dela edo tranpatik dela. Kopia egiten ari den bitartean, ez zaio inori korridorean sartzen uzten. Kopia amaitzen denean, gaizki joan bada, behin eta berriz saiaturiko da zuzen joan arte. Behin kopia egin ondoren, memorian datuak berriz ere *Hashel* helbidetik aurrera gordeko dira, irtendako pertsonen kopurua berriz hasieratuko da eta berriz ere utziko zaie pertsonari sartzen, beti ere banan-banan.

Hurrengoa eskatzen da: Idatzi lengoia algoritmikoan beharrezkoak iruditzen zaizkizun zerbitzu-errutina guztiak eta programa nagusia. Komentatu ariketa ebazteko egiten duzun edozein suposaketa.

-----0000000000-----

10.- Autobide bateko bidesari finkoko zatietako irteeretan dauden ordainlekuetako kabina automatikoak (teleordainketa izenekoak) kontrolatzen dituen sistema bat diseinatzea nahi dugu. Horretarako, irudiko eskeman adierazitako gailuak ditu sistemak, kabina bakoitzean.



Periferikoen kontroladoreen ezaugarriak honako hauek dira:

K_T: Teleordainketa kontroladorea da. Hiru erregistro ditu: kontrol-erregistroa (RKON_KT), egoera-erregistroa (REGO_KT) eta datu-erregistroa (RDAT_KT). Egoera-erregistroak 1 balioa hartuko du teleordainketa sistema duen auto bat detektatzen duenean. Datu-erregistroan —5 bytekoa—, detektatutako autoaren *IZ* identifikazio-zenbakia izango da. Kontrol-erregistroan, *strobe* sekuentzia bat egin behar da; horri esker, egoera-erregistroko balioa Ora igaroko da eta kontroladorea prest geratuko da hurrengo autoa detektatzeko. Kontroladore honen sinkronizazioa **inkesta** bidez egin behar da.

K_Sem: Kabinaren goiko partean dagoen semaforoaren kontroladorea da. Memorian mapeatuta dagoen kontrol-erregistro bat du (RKON_KSem); bertan 1 idazten denean semaforoa gorri jarriko da, 0 idazten denean, berriz, berde.

K_sents: Kabinako irteerako langaren parean dagoen sentsore baten kontroladorea da. Eten bat eskatzen du bere aurretik auto bat erabat pasatu dela detektatzean.

K_DMA: DMA kontroladorea da. Memoriaren eta disko-unitate baten artean transferentziak egiteko erabiltzen da. Honako erregistro hauek ditu:

- **Kontrol-erregistroa (RKON_KDMA):** erregistro honetan lekoa idaztean transferentzia hasiko da. (Automatikoki jartzen da 0an.)

- **Helbide-erregistroa** (RHEL_KDMA): transferituko den blokearen hasierako helbidea.
- **Luzera-erregistroa** (RLUZ_KDMA): transferituko den blokearen luzera bytetan.
- **Egoera-erregistroa** (REGO_KDMA): transferentzia amaitzean, erregistro honek adierazten du transferentzia ondo joan den (1) edo erroreren bat gertatu den (0).

K_DISKO: Diskoaren kontroladorea da. Transferentzia bete ahal izateko kontroladore hau hasieratu behar da. Suposatuko dugu *KdiskoProgramatu()* izeneko errutina dugula hasieratze hori egiteko.

Beste kontroladore guztiak (erlojuarena, teklatuarena eta etenena) ikasgai landutakoak dira. Sistema honetan **teklatuarekiko** sinkronizazioa **etenen** bidez egin behar da. Pantaila memorian mapeatuta dago, PANT_HELB helbidetik aurrera.

Sistemaren **funtzionamendua** honako hau izan beharko da. Kontrolatu beharreko kabinara teleordainketa-sistema duen auto bat iristen denean, kabinaren gaineko semaforoa berde badago, K_T kontroladoreak detektatuko du. Ordainketa automatikoki egingo denez gero (autoaren identifikazio-zenbakiari esker), sistemak kabinako irteeran dagoen langa altxatu behar du, *altxatu_langa()* errutinaren bitartez, autoa pasatzen uzteko. Irteerako langa altxatuta mantenduko da autoa langa azpitik erabat pasatu arte eta 2 segundo gehiago, segurtasun arrazoiengatik, eta orduan jaitziko da, *jaitzi_langa()* errutinaren bitartez. Sistema sinplifikatzeko, suposatuko dugu guztiz beharrezkoa dela langa jaitzi arte itzarotea hurrengo autoa detektatu ahal izateko.

Kobratze automatikoak kudeatzeko, auto bat pasatzen den bakoitzean honako informazio hau gorde behar da memorian: IZ identifikazio-zenbakia —5 bytekoa— eta uneko ordua (*ORDUA* aldagai orokorrean dagoena) —5 bytekoa hori ere—. Informazio hori memorian idazteko, *idatzmem (IZ, ORDUA, autokopurua)* errutina daukagu eta bera arduratzen da zein memoria-helbidetan idatzi behar duen kontrolatzeaz: sistema hasieratzen denean, lehenengo autoari buruzko informazioa INF_HELB helbidetik aurrera idazten du; une horretatik aurrera, kabinatik pasatu diren autoen kopuruaren arabera kalkulatu du helbidea. Orduari dagokionez, honako bi errutina hauek ditugu: *HasieratuOrdua()*, *ORDUA* aldagaiari sistema eragileak esandako uneko balioa esleitzeko, eta *EguneratuOrdua()*, *ORDUA* aldagai orokorra segundo batean gaurkotzen duena.

Kabinaren gaineko semaforoaren kolorea teklatuaren bidez kontrolatzen da:

- G tekla sakatzen denean, semaforoa gorri jarriko da eta ez da onartuko autorik pasatzea kabina horretatik (sistema sinplifikatzearen, suposatuko dugu tekla sakatu behar duen langileak sakatuko duela autorik ez dagoenean). Semaforoaren kolorea aldatzearekin batera, sistemak zera egin behar du: une horretara arte, semaforoa berde egon den bitartean, kabina horretatik pasatu diren autoei buruz memorian metatu duen informazio guztia diskora pasatu behar du, DMA bidez. DMA transferentzia amaitzean, errorerik gertatu baldin bada, beste 2 aldiz gehienez saiatuko da sistema transferentzia errepikatzen, baina 3 saiakeren ondoren transferentzia burutzea lortu ez bada, *errorea_transferentzian()* errutinari dei egingo zaio pantailan mezu bat azaltzeko, eta programari bukaera emango zaio. Errorerik gertatu ezean, funtzionamendu normalak jarraituko du. DMA kontroladorearen sinkronizazioa **etenen bidez** egin behar da.
- B tekla sakatzen denean, semaforoa berde jarri eta funtzionamendu normalari ekingo dio sistemak. Semaforoa berde jarri ondoren pasatuko diren autoei buruzko informazioa berriro INF_HELB helbidetik aurrera gordeko da.

Hurrengoa eskatzen da: Idatzi lengoaia algoritmikoan beharrezkoak iruditzen zaizkizun zerbitzu-errutina guztiak eta programa nagusia. Komentatu ariketa ebazteko egiten duzun edozein suposaketa.