

## Informatika Fakultatea

### Informatika Ingeniaritzako Gradua

#### ▪ Gradu Amaierako Lana ▪

Konputagailuen Ingeniaritza

Etxe-kalte: etxeko komunikazioa errazten duen sare sozial  
parte-hartzaile eta dinamikoa.

---

Jon Mirande

2018 - ekaina



## Informatika Fakultatea

### Informatika Ingeniaritzako Gradua

#### ▪ Gradu Amaierako Lana ▪

Konputagailuen Ingeniaritza

Etxe-kalte: etxeko komunikazioa errazten duen sare sozial parte-hartzaile eta dinamikoa.

---

Jon Mirande

2018 - ekaina

Zuzendaria(k)

Zuzendariaren Izena eta Abizena



# Laburpena

---

Atal honetan proiektuaren laburpena edo nondik norakoa doa. Laburpena denez, nahikoa ohi da bi edo hiru paragrafoekin.

Sistema txertatuak ohikoak dira gaur egun edozein sistema elektronikotan. Hala ere, haien kontsumoa altua da oraindik, eta, ondorioz, bateriak azkar deskargatzen dira. Proiektuan grafenozko transistoreak erabili ditugu kontsumo baxuko sistema txertatu bat diseinatzeko. Betaurrekoetan integratu daiteke, eta lagungarria da errealitate birtualeko inguruak kontrolatzeko.

Hiru algoritmo programatu ditugu: 3Dmatrix, Alphaville, eta guk sortutako E-derra. Emaitzak onak dira hiru kasuetan, baina gure algoritmoa egokiena da ikusmen baxuko egoeretan (estariko, mendian lanbropean).

Hainbat enpresa interesatu dira dagoeneko sistema honetaz eta haren komertzializazioa aztertzen ari gara.



# Gaien Aurkibidea

---

Laburpena .....	iii
Gaien aurkibidea .....	v
Irudi eta taulen zerrenda.....	vii
Algoritmoen zerrenda .....	x
<b>1. Sarrera .....</b>	<b>1</b>
1.1. Zer dira sistema txertatuak? .....	2
1.1.1. Sistema txertatuen historia eta gaurko merkatua.....	4
1.2. Sistema txertatuen historia eta gaurko merkatua.....	3
1.2.1. Lehen esperientziak .....	4
1.2.2. Mikroprozesadoreak, sentsoreak eta adimen artifiziala integratzen duten sistema aurreratuenak: .....	5
<b>2. Proiektuaren helburuak eta baliabideak .....</b>	<b>11</b>
2.1. Helburuak.....	12
2.2. Hardwarea .....	13
2.2.1. GPsa .....	15
2.3. Softwarea: C, Java .....	18
<b>3. Motorrak kontrolatzen duten moduluaren diseinua .....</b>	<b>22</b>
3.1. Abiadura eta azelerazioa.....	23
3.2. Kontrol-algoritmoak.....	26
3.2.1. Sentsoreak .....	31
3.2.1. Motorrak.....	35
<b>4. Konklusioak.....</b>	<b>38</b>
4.1. Egindako lanaren laburpena .....	39
4.2. Merkaturatze-perspektibak .....	45
4.3. Hurrengo pausoak eta hobekuntzak .....	47
<b>Bibliografia.....</b>	<b>50</b>
<b>A Eranskina. Mapak .....</b>	<b>56</b>
<b>B Eranskina. Bezeroaren satisfazioko inkestak .....</b>	<b>99</b>





# Irudi eta Taulen zerrenda

---

## IRUDIAK

1.1. irudia	Exekuzio-urratsen eskema logikoa .....	12
1.2. irudia	Exekuzio-denborak prozesadore kopuruaren arabera, hainbat molekula-tamainatarako.....	15
2.1. irudia	Exekuzio-denborak prozesadore kopuruaren arabera, hainbat molekula-tamainatarako.....	21

## TAULAK

1.1. taula	Sistemaren asmatze-tasak posizioen arabera .....	10
2.1. taula	Salneurriak eta etekinak salmenten eta herrialdeen arabera .....	25
2.2. taula		



# *1*

---

## Sarrera eta hasierako definizioak

1. kapituluaren testua; egokia da hasieran kapituluaren laburpen txiki bat egitea. Esaterako. Kapitulu honetan analizatuko ditugu merkatuan dauden sistema txertatu nagusiak eta haien eragina kontsumoko elektronikan. ....

**1.1. Zer dira sistema txertatuak** Prozesadoreak gero eta azkarragoak badira ere, prozesadore bakar batekin lor daitekeen kalkulu-abiadura ez da behar adinakoa hainbat eta hainbat problema ebazteko [1]. Abiadura handiagoak (bi... sei ordena handiagoak) lortzeko, estrategia desberdina beharko genuke, eta estrategia hori paralelismoa da. Paralelismo hitzarekin honako hau adierazi nahi dugu: programa baten exekuzioa hainbat prozesadoreren artean banatzen da, eta prozesadoreek paraleloan, batera, lan egiten dute.

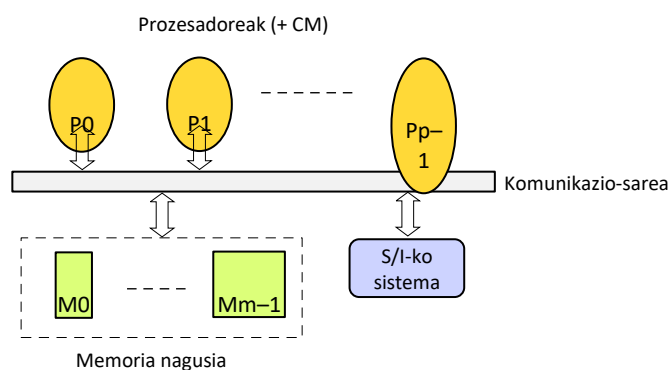
---

Horri ekin baino lehen, zehaztapen bat egin nahi dugu. Arlo bat baino gehiagotan erabil daitezke P prozesadore batera, eskuarki helburu desberdinekin:

- Konputagailu-sareak antolatzeko (LAN, WAN...). P erabiltzailek programa bana exekutatzen dute, batera, baina programa horiek independenteak dira (agian, noizbehinka, datu-transmisioren bat egiten da batetik bestera).
- Hutsegiteekiko tolerantzia lortzeko. Aplikazioen arabera, hainbat modu daude hutsegiteei aurre egiteko..

### 1.1.1. Sistema txertatuen historia eta gaurko merkatua

Memoria partekatuko sistema paraleloetan, prozesadore guztiek partekatzen dute sistemaren memoria osoa; hots, prozesadore guztiek helbide-espazio bera erabiltzen dute: hitz baten helbidea bakarra eta bera da prozesu guztietan. Sistema horien eskema orokorra ageri da 1.1 irudian.



1.1 irudia. Sistema txertatu baten eskema orokorra: memoria, busak, sarrera/irteerako gailuak eta abar.

Hori dela eta, prozesuen arteko komunikazioa oso erraz egin daiteke, aldagai partekatuak erabiliz. Hala, datu bat prozesu batetik beste batera pasatzeko, nahikoa da datu hori memoriako posizio jakin batean idaztea, bigarrenak hortik irakurri ahal izango baitu [2].

#### 1.1.1.1. Ohar pare bat

Memoria partekatuko sistema paraleloetan, prozesadore guztiek partekatzen dute sistemaren memoria osoa; hots, prozesadore guztiek helbide-espazio bera erabiltzen dute: hitz baten helbidea bakarra eta bera da prozesu guztietan.

Hori dela eta, prozesuen arteko komunikazioa oso erraz egin daiteke, aldagai partekatuak erabiliz. 1.1 taulan, esperimenduetan lortutako denborak ageri dira.

Prozesadore kopurua	Exekuzio-denbora (h)	azelerazio-faktorea ( <i>speed-up</i> )	eraginkortasuna
2	24,2	1,8	0,91
4	15,1	3,7	0,87
8	8,4	7,1	0,60
16	5,3	13	
32	4,1	17	

1.1 taula. Exekuzio-denborak prozesadore kopuruaren arabera

## 1.2. Lehenbiziko probak

Diseinuarekin hasi baino lehen, sistema komertzial bat erabili dugu proba batzuk egiteko eta ingurunea kontrolatzean ager daitezkeen arazoak analizatzeko.

Erabili dugun programa nagusia 3 azpiprogramatan banatu da. Hona hemen lehenbizikoa [3].

```
kp = 3;
km = 3;
#pragma omp parallel for shared(kp, km) schedule(static,1)
for (i=4; i<N7; i++)
{
    while (kp < (i-4))
    {
        #pragma omp flush(kp)
    }
}
```

```

}
M[i] = M[i] * 1.7 - P[i-4];
while (km < (i-1))
{
    #pragma omp flush(km)
}
km = km + 1;
#pragma omp flush(km)
R[i] = M[i-4] * 0.9 + R[i];
P[i] = (R[i] - P[i]) / 2.0;
while (kp < (i-1))
{
    #pragma omp flush(kp)
}
kp = kp + 1;
#pragma omp flush(kp)
}

```

1.1 programa. Boot-en algoritmoa

## Bibliografia

---

- [1] Astigarraga A.L. eta Lasarte A.A.: *Embedded systems: past, present, future*. Proc. of the IEEE, v. 17, p. 23., 1999
- [2] Jon Brown: *Berrogei urte eta gero, hau*. [www.ehu.es/jb/hau](http://www.ehu.es/jb/hau)
- [3] *Yo también quiero un Ipod*. Catálogo de Appel, 2013.
- [4]





## A Eranskina: ereduak eta inkestak

---

Kode bitar erabilienak

.....

## ARAU NAGUSIAK

Orriari dagokionez

DIN-A4

Tarteak: goian, ezkerrean eta eskuinean, 3 cm; behean, 2,5 cm

Koadernatzeko tarteak: 0,6 cm

Testu arruntaren motak eta tamainak

times new roman 11, calibri 11, calibri 10, arial 10

lerro artekoa.: sinplea edo 1,15 - 1,2 lerro

paragrafo ondoren: 6 puntu

lehen paragrafoan eskuineko koska gabe

gainontzeko guztietan 0,5 cm-ko koska lehen lerroan

Kodeetarako

courier new 10, courier new 9, courier new 8

Ataletarako

## 1.1 ATAL NAGUSIA

---

### 1.1.1 Azpiatala

### 5.5.5.5 Bestelakoak

Kapituluen hasiera beti bakoitia.

Beti orri zenbatuak.

Taulak: letraren tamaina puntu bat txikiago testuarena baino

Taulen eta irudien oina: letraren tamaina puntu bat txikiago testuarena baino