

Android-entzako SmartWatch prototipo baten
garapena

Unai Urkidi

Irailak, 2015

Laburpena

Gradu Amaierako Lan (GAL) honek Android-entzako SmartWatch baten prototipo txiki bat egitea du helburu. SmartWatch-aren egiturari dagokionez, ondorengo atalak bereizten dira:

- Eskumuturreko erlojua. Arduino plataforman oinarrituta, telefonotik datozen jakinarazpenak pantailatxo batetan adierazten dituen gailu "janzgarria".
- Telefono mugikorra. Android sistema eragilean oinarrituta, hari gabeko teknologia bitartez erlojuari datuak bidaliko dizkion aplikazio bat da.

Eskumuturrekoaren hiru betekizun nagusi ditu:

- Telefonotik bluetooth bidez bidalitako mezuak jasotzea
- Mezu horiek telefonotik eskumuturrekora datu transferentzia egiteko sortutako protokolo baten arabera parseatzea.
- Datuak pantailatxo batean bistaratzea.

Bestalde, Android aplikazioaren funtzio nagusiak ondokoak dira:

- Eskumuturrekoarekin darabilen bluetooth konexioa kudeatzea.
- Sistema eragileak jakinarazpenak kudeatzeko darabilen zerbitzutik jakinarazpenak dinamikoki lortzea.
- Jakinarazpenak (hauen izenburua eta edukia) eskumuturrekoari bidaltzea. Horretarako aplikazio mailako protokolo simple bat sortu da.

Gehigarri bezela, jakinarazpenez gain, Android aplikazioak bateriaren maila ere bidaltzen du.

Edukien Aurkibidea

1	Sarrera	5
1.1	Lanaren Deskribapena	5
1.2	Lan hau aukeratzeko arrazoia	6
2	Hasierako planteamendua	7
2.1	Helburuak	7
2.2	Irismena	7
2.3	Garapenerako tresnak	11
2.3.1	Software Tresnak	11
2.3.2	Hardware tresnak	13
2.4	Denbora plangintza	15
2.4.1	Orduen Kudeaketa	15
2.4.2	Gant Bluetooth	16
2.4.3	Gant Informazio Lorpena	16
2.4.4	Gant Pantaila	17
2.4.5	Gant Interfase Grafikoa	17
2.4.6	Gant Azken Muntaia	18
3	Aurrekarien azterketa	19
3.1	Samsung S Gear azteketa	19
3.2	Moto 360 Azterketa	20
4	Eskakizunen kaptura	21
4.1	Prototipoaren Betebeharra	21
4.1.1	Informazio bidalketa atala	21
4.1.2	Informazio eskuraketa Atala	21
4.1.3	Erabilgarritasunaren Atala	21
4.2	Erabilpen kasua	22
5	Azterketa eta diseinua	23
5.1	Bluetooth diseinua	23
5.2	Notifikazio Zerbitzua	23
5.3	Pantaila	25
5.4	Aplikazio Interfase Grafikoa	26
6	Garapena	27
6.1	Bluetooth Komunikazio Atala	27
6.1.1	Hardware Muntaia I	27
6.1.2	Arduino Programa I	28

6.1.3	Android Aplikazioa I	29
6.2	Pantaila atala	31
6.2.1	Hardware Muntaia II	31
6.2.2	Arduino Programa II	32
6.3	Konexio Kudeaketa atala	33
6.3.1	Android Programa III	33
6.3.2	Hardware Muntaia III	35
7	Egiaztapena eta Ebaluzioa	39
8	Ondorioak eta etorkizuneko lana	42
9	Bibliografia	44

Irudien Aurkibidea

1	Hasierako Planteamendua	8
2	Bluetooth Diseinua	9
6	Arduino UNO	13
7	Rn 44 Bluetooth	14
8	DOG128 Pantaila	14
9	Gant Diagrama Osoa	16
10	Gant Bluetooth Diagrama	16
11	Gant informazio lorpen diagrama	17
12	Gant pantaila diagrama	17
13	Gant interfase grafiko diagrama	18
14	Gant azken hardware muntai diagrama	18
15	Samsung Gear S	19
16	Moto 360	20
17	Erabilpen Kasua	22
18	Bluetooth Klase Diagrama	24
19	Notifikazio Klase Diagrama	25
20	Pantailaren Azkenengo Diseinua	25
22	Bihurgailu Logikoa	27
24	Arduino Serial Monitor I	28
25	Notifikazio egitura	29
26	Notifikazio Textua lortzeko atala	30
27	Notifikazio Aplikazio Izena lortzeko atala	30
28	HCF4050B Bihurgailua	32
29	Pantailaren egitura	33
30	Arduino Pro Mini	35
31	Arduino - Mikrousb bihurgailua	36
32	HC6 Bluetooth	36
33	Mini Pantaila	37
34	Azken protoboard Prototipoa	37

1 Sarrera

1.1 Lanaren Deskribapena

Gradu Amaierako Lan (GAL) honen helburua SmartWatch baten prototipo txiki bat egitea da, prototipo honek SmartWatch baten oinarrizko funtzioak garatu beharko dituelarik. Prototipo hau sistema oso bezela ulertu behar da non funtzio jakin bat betetzen duen software bat diseinatu eta garatzeaz aparte, berarekin lan egingo duen hardware-a ere diseinatu eta garatu egiten da, erabiltzaileari bere jarduera betetzeko behar duen guztia emanaz.

Sistemak bi osagai nagusi ditu: mikrokontroladore batean oinarritutako SmartWatch-a eta Android aplikazio bat. SmartWatch-a eta Android aplikazioaren arteko informazio trukaketa Bluetooth 2.1 bitartez. Egiten da.

Horretarako SPP (Serial Port Profile) profila erabiliko da. Profil honek bluetooth RFCOMM protokoloaren bitartez serial konexio bat emulatzeko aukera ematen du. Mikrokontroladorearen ikuspuntutik SPP erabiltzea oso erreza da, bluetooth moduluarekin duen serial komunikazioaren kudeaketan oinarritzen delako. Android-en ikuspuntutik, SPP erabiltzea erreza da ere, RFCOMM socket bat sortuta, bere gainean SPP zerbitzua erabili nahi dela adierazi behar delarik. Zentzu honetan, RFCOMM protokoloa garraio protokolo bat bailitzan ikusi daiteke, TCP protokoaren antzerako funtzioa betezen duelarik. Horrela, bluetooth moduluak, mikrokontroladorearen eta Android-en artean dagoen hub batek bezela funtzionatzen du.

Serie konexioak, datuen transferentzia egiteko kanal edo hodi bat da. Baina mikrokontroladorean dagoen programak datu ezberdinak bereiztu ditzan aplikazio mailako protokolo txiki bat garatu behar izan da

Informazioa SmartWatch-ean jaso delarik, prototipoan, honek erabiltzaileari jainarazi behar erabiltzaileari gertaera hau jazo dela jakinarazi behar zaio. Hori egiteko seinale ezberdina igorri daitezke (grafikoak, soinudunak, dardarak). Kasu honetan informazioa pantailatxo baten bitartez grafikoki adieraztea erabaki da.

1.2 Lan hau aukeratzeko arrazoa

Idea Bilboko Ingenieritza teknikoko Unibertsitate Eskolako irakasle batek proposatua izan da. Lan honen atal interesgarria gaur egungo teknologia baten garapena da.

Informatikako gradu batean espero den bezela, kurtso ezberdinetan zehar egin ditugun proiektu eta lan gehienak programazio atalean gelditu dira. Horrez gain, gure bizitzean zehar formazio jarraiarako beharko ditugun beste zeharkako kompetentzia garrantzitsu batzuk lortu ditugu, hala nola: arazoen analisia egitea eta konponbideen sintesia bilatzea.

Zentzu honetan, bi arlo hauek batu eta graduan zehar ikasitakoaren erlazioa duen, baina graduko kompetentzia espezifikoetatik arago doan lan edo proiektu bat garatu nahi izan da. Horrela, mugikorreko aplikazio bat (software-arekin erlazioa) eta dispositibo elektronikoko bat (hardware-arekin erlazioa) lotzen dituen lan bat garatu da. Lanak aplikazio praktiko bat izan zezan, Internet of Things izenarekin izendatu izan den industria esparru berrian kokatu nahi izan da, wearable edo janzgarri azpi-eremuan, gaur egungoa diren joera eta teknologien inguruan ezagutza berriak lortzeko.

2 Hasierako planteamendua

2.1 Helburuak

GAL honek Android-entzako SmartWatch baten prototipo txiki bat egitea du helburu. Proiektuaren bidez graduan zehar eskuratutako kompetentzia espezifiko eta zeharkakoak praktikan jarri, eta beste kompetentzi berri batzuk eskuratu nahi dira.

- Proiektu baten kudeaketa eta dokumentazioari loturiko kompetentziak.
- Informazioa bilatzeko eta aurkezteko kompetentziak.
- Teknologia ezezagunen aurrean prestakuntza antolatzeko kompetentziak.
- Dispositibo elektronikoen muntaia eta diseinuari loturiko kompetentziak.
- Android aplikazio baten garapenari eta diseinuari loturiko kompetentziak.

2.2 Irismena

GAL honek, gaur egungo SmartWatch industria esparruari eta berarekin erlazionatutako teknologiei hurbiltzea du helburu. Prototipo baten diseinu eta garapen bitartez, teknologia hauek erabiltzeko kompetentziak lortu nahi dira. Prototipoak SmartWatch baten funtzionalitateak garatuko ditu. Ordua erakutsi beharko du, erloju batek bezala. Android sistema batekin sinkronizatu ahal izango da informazio transferentzia gauzatzeko. Horrez gain informazio hau prozesatu eta pantailartzeko gaitasuna izan behar du.

Ikuspuntu akademiko batetik, proiektu hau errepikagarria izatea bilatzen da, beste edozein ikaslek bertatik ikasteko aukera izan dezan. Horretarako, alde batetik, elektronikazaleen artean eta hezkuntzan arrakasta handia izan duen Open Source Hardware (OSHW) ekimenaren inguruan ekoizten diren elektronika modulo merkeak erabiliko dira. Bestetik, software plataforma irekiak erabiliko dira.

Lan edo proiektu honen garapena antolatzeko, arazoa sistema osotuko duten osagaien arabera zatitu da: SmartWatch-a alde batetik eta Android aplikazioa bestetik.

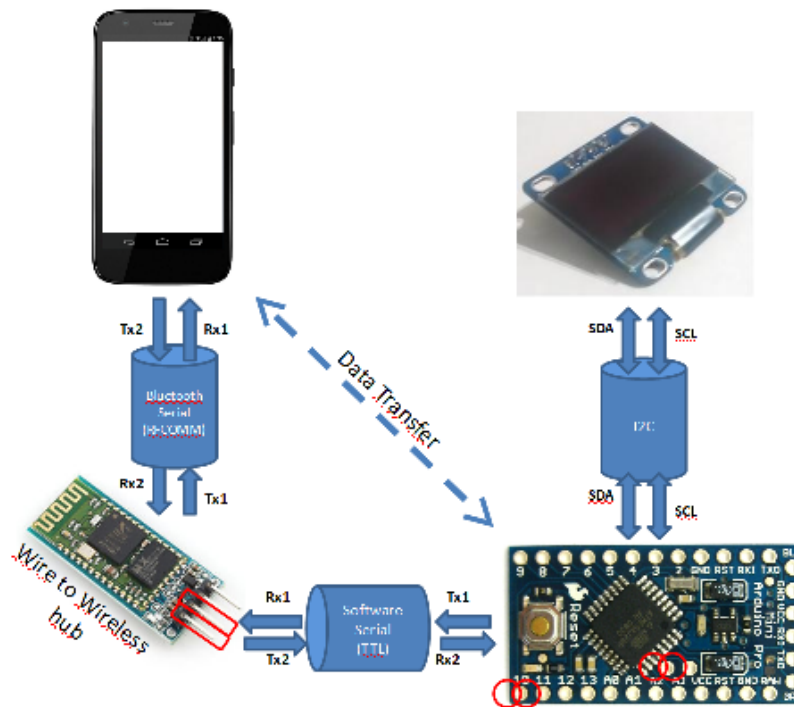


Figure 1: Hasierako Planteamendua

Smart Watch-aren kasuan (elektronikako zatia) beste azpianolakuntzabat jarraitu da. Izan ere, kontuan izan behar da ikaslea ez zela mikrokontroladore baten programazioan aditua, ezta hardware-aren behe mailako funtzionamendu eta konexio moduetan. Beraz, arlo honetan, prestakuntzari dagokionez hutsetik hasi behar izan da. Horrela, lehenengo fase batean, ez da tamaina kontuan hartu hardware-a aukeratzeko orduan, lan egiteko erraztasuna eta programazio bitartez funtzionalitate nagusia martxan jartzea helburu nagusiak izan direlarik. Bigarren fase batean, eskumuturreko erloju baten kutxan sartu daitezkeen hardware modulo txikiak aukeratu dira, kodea zuzenean berrerabili ahal izan delarik.

Honekin atal bakoitzaren diseinu eta garapena bereiziz, kudeaketa eta mantenimendu lanak erraztea bilatu da. Bereizketa, hau izanik. (ikus. 1)

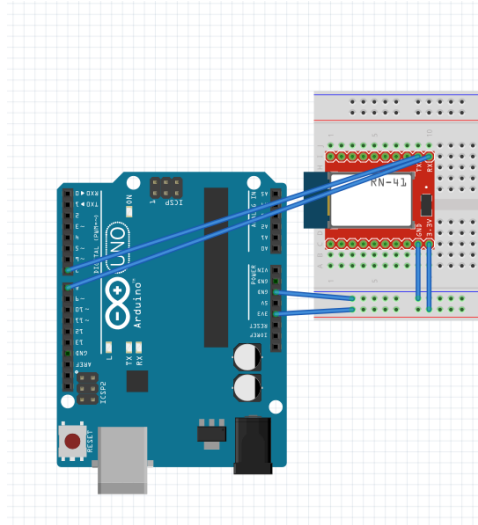


Figure 2: Bluetooth Diseinua

AurreProiektua: Proiektua hasi baino leen, argi izan beharrezko puntuen definitzea. Puntu honetan beste atalen hasierako plangintza egingo da. Garapena egin ahal izateko erabili beharrezko tresnak aukeratuko dira. Eta tresna hauek lan egiteko beharrezko instalazio eta konfigurazioak egingo dira.

Bluetooth Konexioa: Helburua, Android sistemaren eta prototipoaren arteko Bluetooth konexioa gauzatzea da. Konexio hau Bluetooth Komponente baten bitartez gauzatuko da. Hasierako ideia, (ikus. 1 Android Bluetooth konponen transferentzia, RF-COMM socket bidez egitea da. Hasieran prototipoa malgua izango da, hau da protoboard batean muntaia egingo da amaierako microhardwarearen antzeko pieza handiagoak erabiliz. (ikus. 2). Hau lortzeko hurrengo pausoak jarraituko dira.

- Beharrezko Hardware konponenteen analisia egin.
- Prototipoaren hardware konponente handien diseinua egin.
- Bluetooth konponentea prototipoan integratu.
- Arduino softwarea garatu.
- Android sistemaren softwarea diseinatu.

- Android sistemaren softwarea garatu.
- Probatu funtzionalitatea.

Informazio Lorpena: Android Sistematik, pantailaratu nahi den informazioaren lorpena dugu helmuga. Hasierako planteamendua (ikus. 1) jarraituz, behin Android-eko datuak eskuratuta, mikrokontroladoreak, Bluetooth-etik jasotako informazioa interpretatu behar du. Bluetooth mikrokontroladore informazio transferentzia serial konexio bat emulatuz lortuko da. Informazioa karakterez karaktere irakurriko denez, aplikazio mailako protokolo bat garatu beharko da. Hau lortzeko pausoa hauek dira:

- Beharrezko software tresnen analisia.
- Tresna hauek erabiltzeko diseinua.
- Android Sistemaren softwarea garatu.
- Arduino protokoloa diseinatu.
- Probatu funtzionalitatea.
- Aurreko helburuari integratu.

Informazioa Pantailaratzea: Lortu beharrekoa, aurreko pausuetan bidaltzen, eta prozesatzen den informazioaren pantailaratzea da. U8glib[1] liburutegia erabiliko da pantailaren garapenerako. Honen aukeraketa, pantaila handia eta txikiarekiko konpatibilitatean datza. Liburutegiari esker bi pantailek kode bera inplementatuko dute eta migrazioa lerro baten ordezkaketan datza. Lortutakoa onartzeko pausu hauek beteko dira:

- Beharrezko Hardware konponenteen analisia egin.
- Konponente-prototipo integrazioa diseinatu.
- Modulua konektatu.
- Aurreko atalekin zuzen lan egiten duela baieztatu.

Aplikazioaren Interfase Grafikoa: Erabiltzaileak prototipoa, aplikazioaren bitartez, kudeatzea da helburua. Pauso hauek jarraituko dira helburua lortzeko:

- Prototipoarekiko erabiltzaileen beharrak aztertu.
- Behar horiek betetzen dituen interfase grafiko bat diseinatu.
- Interfase grafiko hori garatu.
- Aztertutako beharrak asetzen dituela baieztatu.

Hardware azken muntaia: Atal honetan, aurreko funtzionalitateak betetzen dituen beste prototipo bat sortuko da. Prototipo honek eskumuturrean eraman ahal izateko tamaina izan behar du. Puntu hauek beteko ditu zati honen garapenak:

- Prototipoaren hardware konponente txikien diseinua egin..
- Diseinua jarraituz prototipoa garatu.
- Balioztatu prototipo berriak, prototipo handiaren funtzionalitateak betetzen dituela.

2.3 Garapenerako tresnak

2.3.1 Software Tresnak

Garapen Tresnak

Android Studio: Smartphon-eko aplikazioaren garapenerako programazio ingurumena.(ikus. 3a)



(a) Android Studio



(b) Arduino Software



(c) Bluetooth Terminal

Arduino Software: Prototipoaren garapenerako programazio ingurumena.(ikus. 3b)

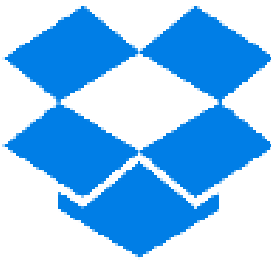
Bluetooth Terminal[7]: Bluetooth integrazioa zuzena dela frogatzeko Android softwarea.(ikus. 3c)

Backup Tresnak

GitHub: Kodearen irudiak egiten ditu. Igotako edozein egoerara itzultzeko aukera ematen du. Egora bakoitzak bere izenburua eta deskribapena darama, kodearen zein atal aldatu den ere ikus daiteke.(ikus. 4a)



(a) GitHub



(b) Dropbox

Dropbox: Informazioa interneten gordetzeko tresna.(ikus. 4b)

Dokumentazio Tresnak



(a) Cacoo



(b) Visual paradigm



(c) Overleaf

Cocoo: Android aplikazio diseinuak egiteko online tresna.(ikus. 5a)

Visual Paradigm: Aplikazioaren klase diagrama zein Sekuentzia diagramak egiteko tresna.(ikus. 5b)

Overleaf: Dokumentazioa egiteko, LaTeX-en oinarritzen den on-line tresna.(ikus. 5c)

2.3.2 Hardware tresnak

Prototipoaren probak egin ahal izateko amaierako konexioak egin gabe honako konponenteak erabiliko dira.

Arduinoz aparte badaude Internet of Things arloan erabili daitezkeen beste plataforma batzuk, horien artean Raspberry Pi eta Beaglebone ezagunenak direlarik. Arduino sistema eragilerik gabeko mikrokontroladore baten oinarritzen den bitartean, Raspberry Pi eta Beaglebone-k Linux bertsio berezi bat exekutatzeko duen SoC (System-on-Chip) bat dute, bere CPU, GPU eta memoriarekin, honek dituen abantaila guztiak. Hala ere, Raspberry Pi-a eta Beaglebone-a ez daude GAL honetan darabilgun eskumuturreko bezala sistema txertatu batean sartuak izateko eta modu iraunkorrean lan egiteko aukerarik. Arduinok, ostera, baditu eskumuturrekoaren tamaina eta kontsumora egokitzeko plaka ezberdinak.

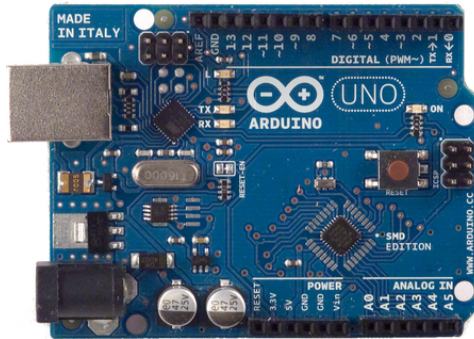


Figure 6: Arduino UNO

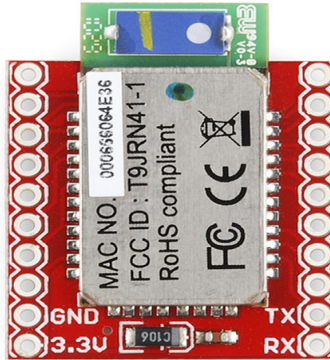


Figure 7: Rn 44 Bluetooth

Arduino UNO: Prototipoaren informazioa prozesaketaz arduratuko da. Prototipoaren software garapena konponente honen gainean izango da. Prototipoaren bihotza dela esan dezakegu.(ikus. 6)

RN 44 Bluetooth: Konexioak egin ahal izateko Bluetooth konponente baten beharra dugu. Honek bluetooth konexioa ahalbidetuko du Android Sistemaren eta Arduinoaren artean.(ikus. 7)



Figure 8: DOG128 Pantaila

DOG M128 Pataila: Pantaila hau Arduinoak Bidalitako informazioa pantailaratuko du. Pantaila grafiko bat denez aurretiazko software garapen bat dakar erabiltzeak (ikus. 8)

2.4 Denbora plangintza

2.4.1 Orduen Kudeaketa

Garapenak	Denbora (Orduak)
Dokumentazioa	40
Arduino ezagutzak lortzea	10
Android ezagutzak lortzea	25
Latex ezagutzak lortzea	5
Merkatu analisia	10
Bluetooth Konponenteen azterketa	5
Bluetooth diseinua	15
Bluetooth hardware muntaia	3
Bluetooth Arduino garapena	30
Bluetooth Android garapena	50
Bluetooth probak	7
Notifikazio tresnen analisia	5
Notifikazio atal diseinua	10
Notifikazio atal Android garapena	35
Notifikazio atal probak	5
notifikazio integrazio probak	2
Pantaila Konponente analisia	5
Pantaila hardware diseinua	5
Pantaila Arduino garapena	10
Pantaila integrazio probak	1
Erabiltzaile beharren analisia	2
Interfase diseinua	4
Interfase grafiko garapena	10
Hardware eskumutur diseinua	3
Hardware eskumutur garapena	5
Hardware eskumutur balioztapena	1
Guztira	303

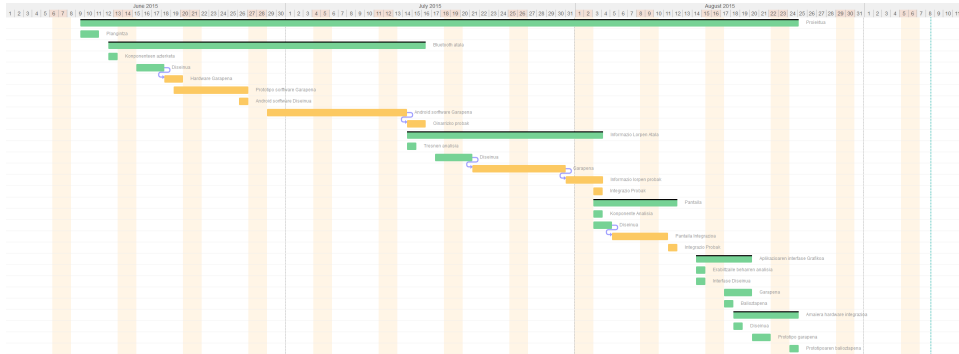


Figure 9: Gantt Diagrama Osoa

Ondoren Proiektuaren denboraren banaketa ikusteko Gannt bat garatu da.(ikus. 9) Gannt-a 5 ataletan banatzen da.

2.4.2 Gant Bluetooth

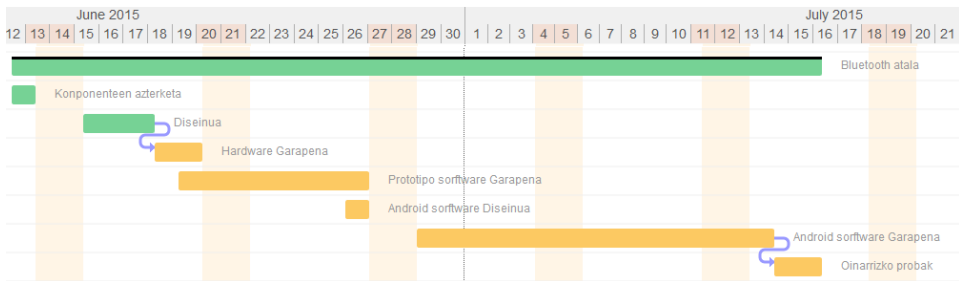


Figure 10: Gant Bluetooth Diagrama

Atal honi (ikus. 10), Bluetooth-aren garapena dagokio. Bluetooth, konexioen diseinua, tresnen ikasketa, software garapena da atal honen helburua.

2.4.3 Gant Informazio Lorpene

(ikus. 11) ikusten den atalean, Android sisteman, software bitartez, informazioaren lorpenari zuzendua dago.

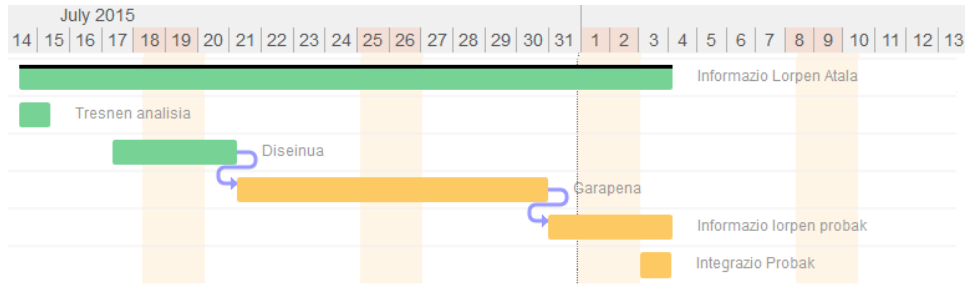


Figure 11: Gant informazio lorpen diagrama

2.4.4 Gant Pantaila

Pantailaren garapenerako denboraren banaketa isladaten du gant honek.(ikus. 12) Esleitutako denbora honetan pantailaren integrazioa egin eta funtzionamendua zuzena dela bermatu beharko da.

2.4.5 Gant Interfase Grafikoa

Prototipoa aurreko puntuak egiteko gai denean, aurreko funtzionalitateak kudeatzeko, interfase grafiko bat garatuko da Android-en. Lan horretarako esleitua dagoen denbora grafikoan (ikus. 13) isladatzen da.

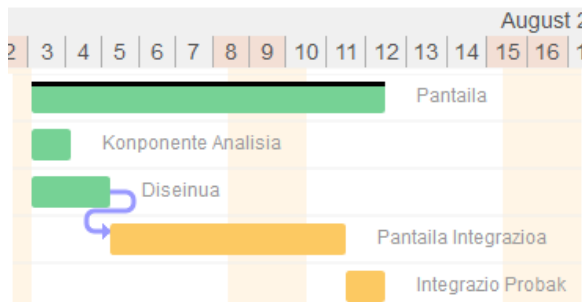


Figure 12: Gant pantaila diagrama

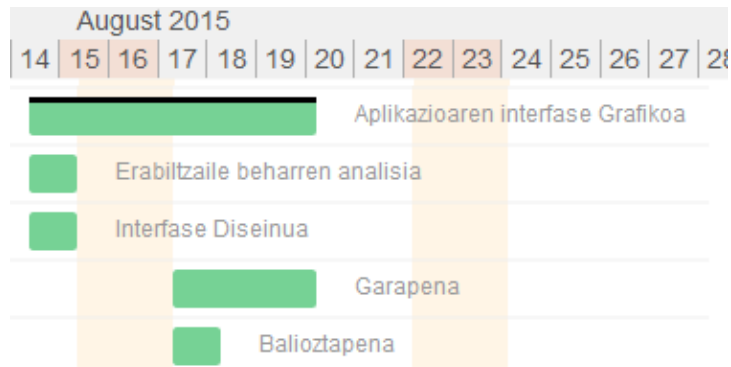


Figure 13: Gant interfase grafiko diagrama

2.4.6 Gant Azken Muntai

Prototipoa konponente handiekin balioztatua dagoenean, diagraman (ikus. 14) ikus daitekeen denboran, prototipo txikia egitea kalkulatzen da.

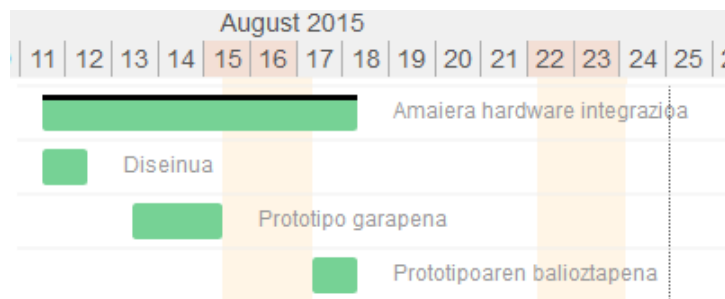


Figure 14: Gant azken hardware muntai diagrama

3 Aurrekarien azterketa

Gaur egun hainbat SmartWatch salgai daude, zein eboluzio egoeran dagoen merkaturak jakiteko hauek zer egiteko gai diren aztertu behar dugu. Non dagoen funtzionalitate mugaren ideia izateko, Samsung eta Motorola-ren smartwatch salduenetariko[10] datuak hartuko ditugu. 2014ko datuak erabiliko dira urte osoko salmenten datuak eskuragarri daudelako. Samsung eta Motorola dira aukeratuak enpresak, smartwatch merkatuaren zati handienak dituztenak zirelako 2014-an(gehitu biblioteka). Berez onenak izan ez daitezke baina gehien saltzen direnak dira. Aukeratuak Samsung Gear S (ikus. 15) eta Motorola Moto360 (ikus. 16) ren funtzionalitateak hartuko ditugu.

3.1 Samsung S Gear azterketa

Lehendabizi Samsung gear S-aren funtzionalitateak aztertu ditugu. SmartWatch hau 2014 urteko salduena[10] izan da, 1.2 milioi unitateekin.

Funtzionalitateen artean, dituen hainbat sentsoreen irakurmena dugu: Azelerometroa, giroskopio, konpasa eta baita pulsu irakurle bat ere. Sentsoreak alde batera utzita, Bluetooth, WiFi eta GPS ditugu. Mezuak irakurtzeko deiak pantailaratzeko eta baita irudiak ikusteko zein musika erreproduzitzeko ahalmena du. Prezioa 350\$.[8]



Figure 15: Samsung Gear S

Beste aldean Merkeagoa eta sinpleagoa den Moto 360-a daukagu. Honen salmentak 0,5 milioikoak izan ziren[10]. Samsung gear S-arekin alderatuz, erdia baino gutxiago saldu zuen.

3.2 Moto 360 Azterketa



Figure 16: Moto 360

Moto 360 aren funtzionalitate nabarmenenak ez dira asko ezberdintzen Samsung eko modeloarengaitik. Biek informazioa lortzen dute Android sistema batetik, biek dituzte oinarrizko sentzore berdinak. [5]

Ezberdinasun nasgusieneren arten hauek ditugu. Moto 360-ak android wear sistema du, Samsung gear S ordea Tizen dabil. Moto 360-ak ez du ez Samsung duen 3G erabiltzeko gaitasuna, ez idazteko teklatu birtuala. Bestalde Moto-aren eskumuturrekoak aldatu ahal dira, eta email osoa irakurtzeko

gaitasuna du. Azkenik, Moto-aren prezioa 250\$[8] 100\$ merkeagoa.

Merkatuan zer dagoen aztertuz, lortu ditzakegu zeintzuk diren falta zaizkien funtzionalitateak.

Argi dago SmartWatch hauek onenak izan direla, eta merkatua gora egiten ari dela[9][4], beraz pertsona gehiago asetzea izan behar dugu buruan. Pertsona gehiago asetzeko ideia buruan izanda, kontuan hartu behar dugu, sistema hauek, Samsung Motorola baino gehiago, itxiak direla. Hau da, ezin dira guztiz aldatu ez hobetu.

Moto 360-ak, Samsung-ak ez bezala, kodea zabaldua du[6] hau da, edonork alda dezake nahi duen bezala bere sistema. Baina modularitateari dagokionez ez bata ez besteak ez du behar hori asetzen. Sistema osoa ordaindu behar da, nahiz eta atal batzuk hainbeste ez erabili. Gainera atal batek zure beharrak asetzen ez baditu, edo apurtzekotan, ezin da aldatu.

4 Eskakizunen kaptura

Lan hau smartphoneko erabiltzaile arruntei zuzendua dago batez ere. Honi esker mugikorraren informazioa eskumuturrean irakurri ahal izango dute. Erabiltzailearen esperientzia hobetzeko, aplikazioan erabilera erraztasuna da garrantzia handienetarikoa duen atala.

4.1 Prototipoaren Betebeharra

Aplikazioak hainbat puntu behar ditu, zati desberdinen barruan. Denek batera eta zuzen lan egitea da betebeharr nagusia.

4.1.1 Informazio bidalketa atala

Ziurtatu behar da, informazioa, emandako debora tartean prototipora heltzea. Dei batean informazioa galtzea onargarria da, baina ezin dena gertatu, komunikazio portua blokeatzea da. Kontuan hartu behar da ere, bidalitako informazioa, egitura egokia dararamala, eta prototipora heltzen denean honek zuzen interpretatu ahal duela.

Konexioa, erabiltzaileak zerbitzua piztu ostean bakarrik gertatuko da, eta frekuentzia aldatzerakoan frekuentzia tarte hori errespetatu egin beharko da.

4.1.2 Informazio eskuraketa Atala

Eskuratutako informazioa beti gordeko da. Bakarrik beste datu bat datorrenean ezabatu egingo da. Nahiz eta aplikazioak huts egin bidali baino lehen, datua gorde bada, hurrengo seinalean aplikazioak datua eskuragarri izan behar du.

Aplikazioak, behar duen informazioa bakarrik prozesatuko du. Behar ez den informazioa ez da log batean ezta datuetan gordeko, honen beharrik ez baitago.

4.1.3 Erabilgarritasunaren Atala

Aplikazioak hainbat tresna izan behar ditu eskuragarri. Tresna horiek, deien kudeaketa tresnak, notifikazio proba sortzea, eta

bluetooth gailuen kudeaketa dira.

Zerbitzuaren deien kudeaketarako trenak, hau piztea, itzaltzea eta frekuentzia aldatzea, eskuragari izan behar ditu erabiltzaileak. Bluetooth gailuei dagokionez, aplikazioa bluetooth ezberdinetara konektatzeko ahalbidea izan behar du era erraz batean.

4.2 Erabilpen kasua

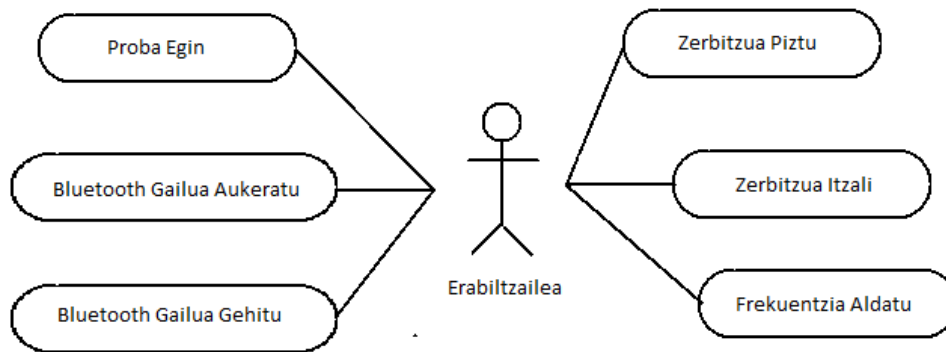


Figure 17: Erabilpen Kasua

Aplikazioa berez, erabiltzailearen ikuspegitik, konfigurazio tresna bat besterik ez da, horregaitik ez du interakzio askorik egin behar zerbitzua zuzen ibiltzea lortzeko.

Proba egin: Kasu honetan notifikazio bat sortuko da. Zerbitzua piztuta badago, honek datua hartuko du eta hurrengo deian bidaliko du informazioa.

Bluetooth Gailua Aukeratu: Erabiltzaileari leiho bat erakutsiko zaio. Hemen Android sistemak jada sinkronizatuta dituen gailuak agertuko dira. Hemen behin gailua aukeratua informazioa gorde egingo da zerbitzua piztean konexioa gauzatzeko.

Bluetooth Gailua Gehitu: Erabiltzaileari leiho bat erakutsiko zaio. Hemen, Android sistemaren bluetooth leihoa agertuko da, non bluetooth gailu berriak gehitu daitezke lehen aipatutako zerrendari.

Zerbitzua Piztu: Zerbitzua hasi egingo da, lehenengo deia, datuak badaude, 5 segundura egingo da, ondoren 5 minuturoko frekuentzia dago.

Zerbitzua Itzali: Zerbitzuaren frekuentzia deiak itzali egingo dira, ez da dei gehiagorik egingo.

Frekuentzia Aldatu: Zerbitzuaren dei frekuentzia aldatu egiten da. Frekuentzia, sartutako minutuetara aldatuko da. Frekuentzia aldatu eta 5 segundu geroago dei bat egiten da.

5 Azterketa eta diseinua

Proiektu honen antolaketa lau zatitan banatzen da. Bluetooth-a, notifikazio zerbitzua, pantaila eta aplikazio interfase grafikoa. Zati bakoitza besteekiko indpendientea egitea da ideia nagusia, horrela arazoren bat egotekotan, modulo bat bakarrik konpondu beharko litzateke.

5.1 Bluetooth diseinua

Hasiera batean Bluetooth kudeatzailea bi klaseetan banatuko ditugu. Lehenengo klasea, Bluetootharen konexio saiakeraz arduratuko da, behin konexioa lortua, bigarren klasea izango da informazioa bidaliko duena (ikus. 18).

Klase egitura hau Android Developers-eko tresnak erabiltzeko zuzendua dago. Tresna hauek arazoak ematen dituzenez gure aplikazioan, diseinu hau baztertua izan da. BluetoothSPP liburutegia[2] da azken bertsioan erabiltzen den tresna.

5.2 Notifikazio Zerbitzua

Notifikazio zerbitzuari dagokionez, hasierako diseinuan komunikazio zuzena egiten zen notifikazio zerbitzua eta bluetooth-aren artean (ikus. 19). Beraz notifikazio bat jasotzean honen bidalketa automatikoa suertatzen zen.

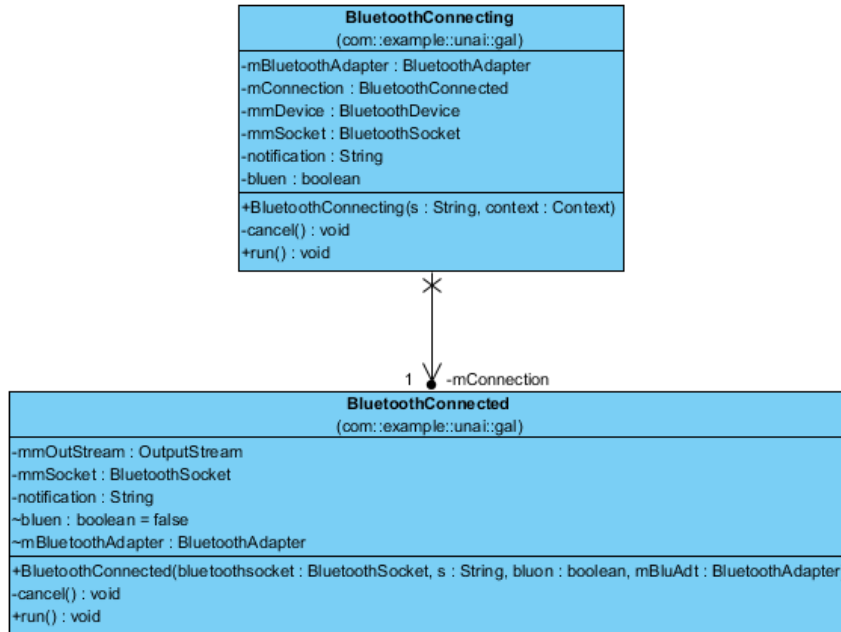


Figure 18: Bluetooth Klase Diagrama

Lehen aipatu bezala Bluetooth-ak arazoak sortzen zituen, baina ez zen bakarra. Notifikazio zerbitzuaren hasierako diseinu honek arazoak zekartzan lehenengo frogak egin eta gero. Bluetooth-Alarma diseinu berriarekin Bluetooth-blokeo arazoak konpondu egin dira.

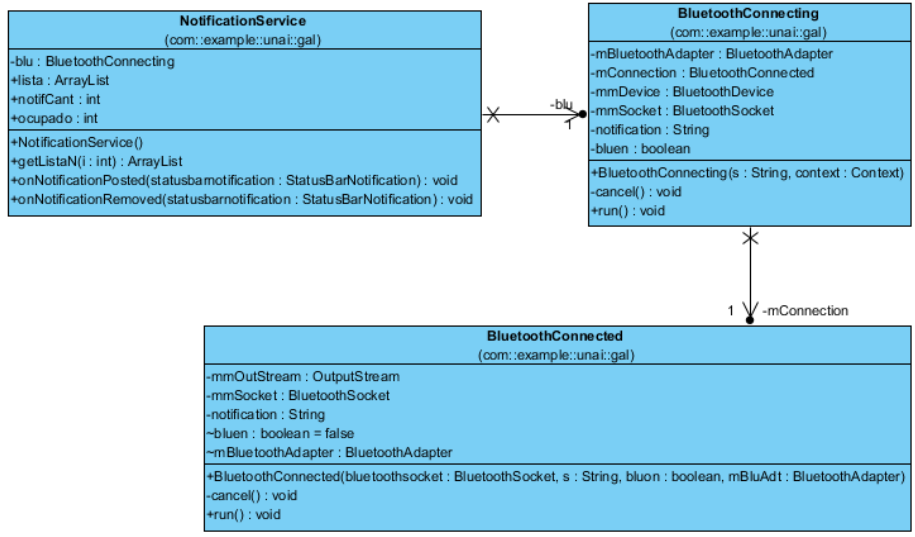


Figure 19: Notifikazio Klase Diagrama

5.3 Pantaila



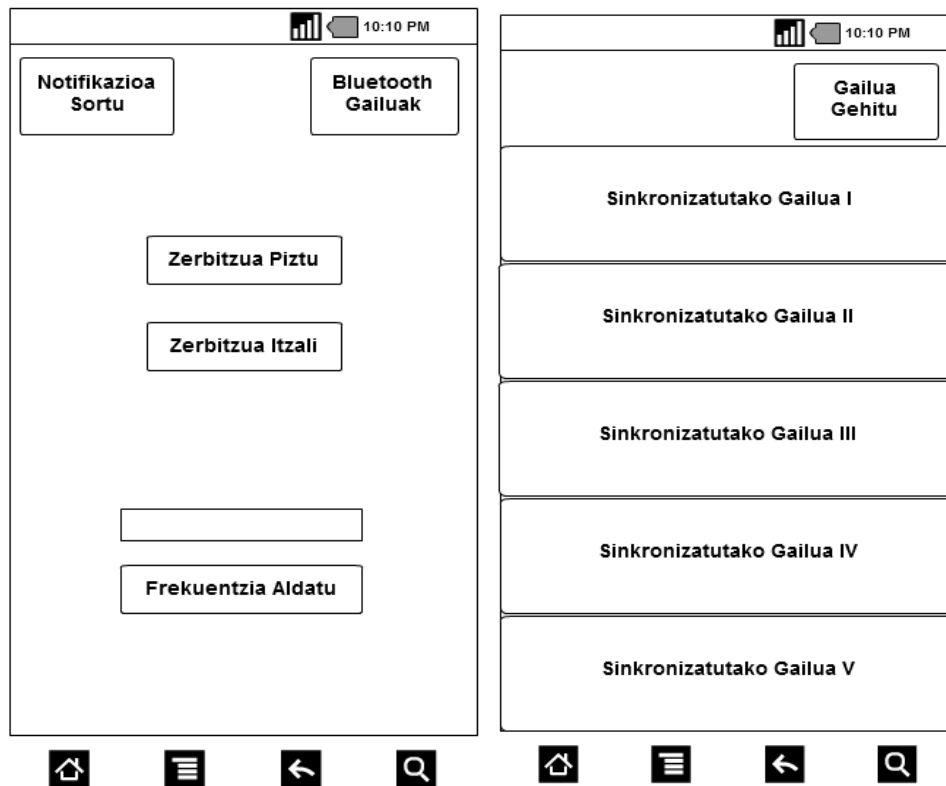
Figure 20: Pantailaren Azkenengo Diseinua

Pantailak arazo bat zekarren. Ideia eskumuturrean eramatea denez, pantaila ezin da oso handia izan, beraz, kontuz aukeratu da non

erakutsi zein informazio. Hasierako planteamenduak ez zekarren notifikazio kantitate datua, erabiltzaile baten hobekuntza da. Diseinu hau da amaieran mantentzen dena.(ikus 20)

5.4 Aplikazio Interfase Grafikoa

Interfase grafikoaren diseinua egiterakoan kontuan hartu egin da, konfigurazio leihoak direla. Era ulergarri eta erraz batean azaldu egin behar dira, erabiltzaileari eskuragarri utzitako tresnak. Leiho nagusian, (ikus. 21a) konfigurazio tresna beharrezkoenak azaltzen dira, gehienetan erabiliko direnak. Bigarren leihoan ordea, (ikus. 21b) maiztasun txikiago duen konfigurazio tresna jarri da. Nahiz eta azken hau haienbeste ez erabili, gailuen arteko konfigurazio aldaketa erraz egiteko pentsatua da diseinua.



(a) Leiho Nagusia

(b) Bluetooth Gailu Leihoa

6 Garapena

Proiektuaren eboluzioa bere barne funtzioetan eta hardware tamainan oinarritzen dira. Garapen hau hiru atal ezberdinetan garatu da. Lehenena Bluetootharen muntaia eta honen hasierako bertsioa da. Bigarren atalean Informazio hori pantailarekin bateratzean datza. Azkenik, informazioaren kudeaketa eta muntaia dago.

6.1 Bluetooth Komunikazio Atala

6.1.1 Hardware Muntaia I

Bluetooth-aren konexioa Arduinora, da lehendabizi lortu behar dena. Bluetooth-ak serie protokoloa erabiltzen du komunikatzeko. Arduino UNO-ak serie komunikaziorako portu bakarra du, eta horren beharra dugu Arduino-Konputagailu konexioa gauzatzeko. Arazo honen konponbidea, SoftwareSerial[3] liburutegia da. Liburutegiari esker, beste pin pare bat hautatzen dira serial portu bat emulatzeko. Pare honek serial konexio bat bezala lan egiten dute.

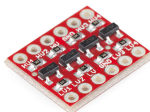


Figure 22: Bihurgailu Logikoa

Hemen agertzen da lehen arazoa. Bluetooth-ak 3,3V-etan bidaltzen eta irakurtzen du informazioa, beraz nahiz eta Arduinoak 3,3V-ko irteera izan, pin digitalen seinalea 5V-tan dago. Hau da, arduinoak Bluetooth-a piztu ahal du baina informazio transferentzia egiterakoak, Arduinoak Bluetooth-a erretzen amaituko luke. Horri aurre egiteko bihurgailu logiko bat inplantatu da (ikus. 22).

Behin konexioa gauzatzea lortu denean, hauen arteko konexioa badabilela zihurtatu behar da. Horretarako bi tresta erabiltzen dira. Lehenengoa, Bluetooth Terminal[7], Android sistematik informazioa bidaltzeaz arduratzen da (ikus. 23a eta 23b), eta bigarrena, Arduino Serial Monitor(ikus. 24), Arduino software

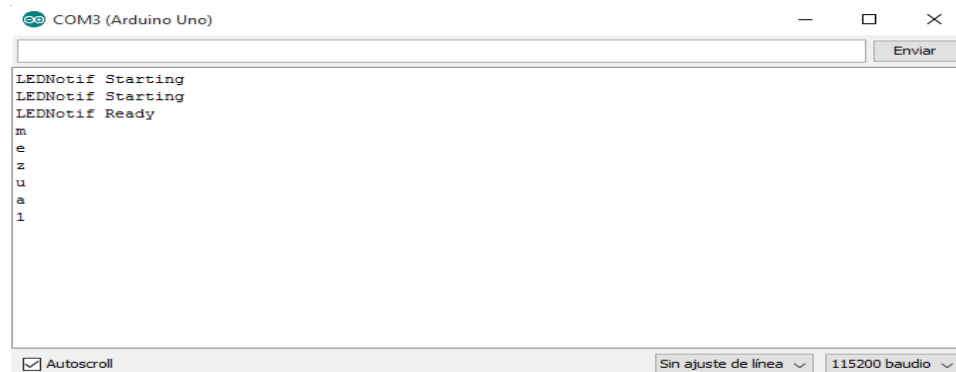
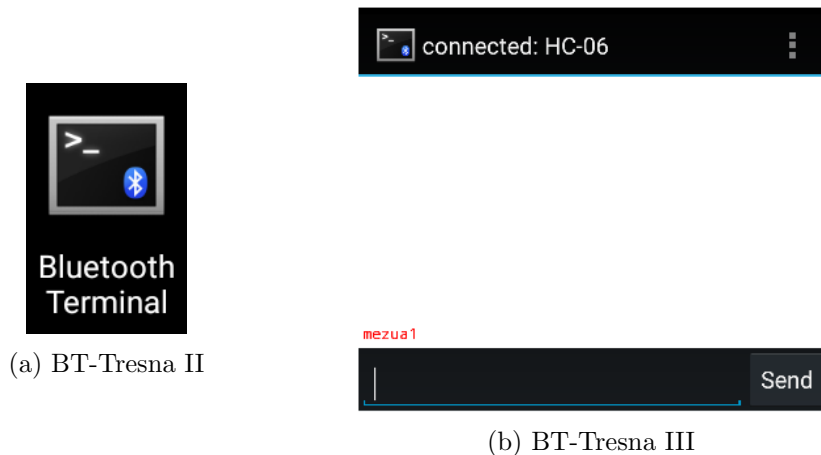


Figure 24: Arduino Serial Monitor I

log-ak pantailaratzen ditu. Kasu honetan, aukeratutako datuak, Bluetooth-etik irakurtzen direnak izango dira.

6.1.2 Arduino Programa I

Hardware atala antolatua eta balioztatua, Arduinoaren kodea da ondorengo helmuga. Hasierako kasu honetan, bi helmuga lortu nahi ditugu. Arduinoak irakutzen dituen datuak inprimatzea, eta datu horiek gordetzeko gai izatea.

Pauso errezena informazioaren pantailaratzean dago, egin beharreko gauza bakarra log baten bidez irakutzen dugun datua pantailaratzean datza. Ondoren, datuen zatiketa eta irakurketan hasi

baino lehen aurreplanteamendu bat dago, datuak nola bidako direnaren planteamendua.

Datuak karakterez karaktere irakurriko dira, beraz karaktere berezi batzuk erabiliko dira jakiteko non amaitzen den datu bat, eta non hasi egiten den hurrengoa(ikus.25).

(1): *Notifikazio bidali duen aplikazioa.* (2): *Mezua.*
(3): *Notifikazio kopurua.* (4): *Bateria portzentaia.*
(5): *Android-eko Ordua.* (6): *Android-eko Minutuak*

@1—2 #3#4%5&6&

Figure 25: Notifikazio egitura

Sistema itxaron egoeran dago lehenengo karaktere berezia irakurri arte. Datu hau heltzen denean , irakurketa egoreara igarotzen da. Hemen kontuan hartzen da, irakurtzen ditugun datuen atal bat, pertsona batek bidalia izan daitekeela. Hori jakinda, estimatu da, atal horretan bi karaktere berezi erabiltzea arazorik ez izateko. Arduinoari datu guztiak lortu direla jakinarazteko azkenengo karaktere berezi bat definitzen da, Arduinoa hau irakurtzean, datuak gorde, eta zikloa berrabiarazten du. Zikloa amaitu baino lehen log bat dago, honek lortutako informazio guztia, prozesatu ostean, zuzen gorde dela ziurtatzen du.

Informazioaren irakurketa zuzena dela egiaztatzeko, eskuz bidali egin izan da Bluetooth-Terminal erabiliz. Bidalitako informazioa, lehen aipatutako Arduino Serial monitor bidez, aztertu da, eta amaierako log-ari esker datuen prozesamendua balioztatu da. Gainera, Android-eko bluetooth probak errazteko, Arduinoari kode zati bat gehitu zaio, non zikloa amaitzean, led berde bat pizten den.

6.1.3 Android Aplikazioa I

Androideko aplikazioa programatu baino lehen, probak egiteko ingurumen bat zehaztu behar da. Ingurumen hori mugikorra izatea hautatu da, azken erabiltzailearen esperientziarekiko bardintasun gehiena duen sistema delako.

```
Notifikazioa.extras.getCharSequence("android.text").toString()
```

Figure 26: Notifikazio Textua lortzeko atala

Notifikazioen atala egiterakoan, kontuan hartu da zeintzuk datu nahi diren. Horretarako notifikazioen egitura, eta zerbitzuaren funtzionamendua jakin behar dira. Android Developers-ko Notification Listener Service da hori lortzeko aukeratutako tresna. Bertan, zerbitzua inplementatzeko beharrezko ia informazio guztia dago. Falta den datu bakarra, notifikazio lerroko datuen egitura da. Arazoa, lerro luzeegi batetik, nahi dugun informazioa hartzean datza, eta aurre egiteko saiakera simple batzuk egin dira.

```
appName = appInfo.processName;
appNameBundle = appName.split("\\.");
appName = appNameBundle[appNameBundle.length-1];
```

Figure 27: Notifikazio Aplikazio Izena lortzeko atala

Aplikazioak, notifikazioak irakurri ahal izateko, zerbitzu honetarako baimenak eman zaizkio. Datuetatik informazioa lortzeko, hauek log batean pantailaratu dira. Pantailaratzetik egitura ulermena lortu da, eta berarekin datuak lortzeko gaitasuna. Bi datu dira beharrezkoak, zein aplikazio den notifikazioa bidali duena, eta honen mezua. Hau da lerrotik lortzen den lehenengo datua (ikus. 26). Aplikazio izena lortzeko hainbat froga egin ostean lortzeko kodea garatzea lotru da. Izen osoa konposatua da, beraz bidali baino lehen aurretratatamendu bat egiten da, izen simplea lortuz. (ikus. 27).

Informazioa eskuratu denean, Bluetooth Konexio bat sortzen da. Bluetooth-ak bi zati ditu, konexioa sortzea eta Informazioa bidaltzea. Bigarren atala ezin da suertatu lehenengoak bere lana egin ez badu. Bluetooth-a konektatzen duen atalak, behin bere helburua lortua, Bluetooth bidaltzaileari informazio socket-a, bidaliko dio. Azken hau informazioa bidaltzeaz bakarrik arduratuko da. Kontuan hartu da, aplikazioaren ezintazuna bluetootharekin lan egiteko baimenik gabe, notifikazioen antzera, Aplikazioari baimenak eman zaizkio

Konexioa sortarazteko hasierako prototipo honetan, Bluetooth-a aktibatzen da, lehendik ez bazegoen, eta eskuz sartutako MAC helbidera konexioa zortzen da. Aktibatu egiteko, Bluetooth-ak Sistemaren bluetooth gailuaren kontrola hartzen du. Konexioa sortzeko, MAC baten beharra du, hau Bluetooth plakatik lortu izan da probak egiteko. Behin konexioa lortua, honek informazioa bidali behar du, horretarako beste thread baten daukagun Bluetooth bidaltzaileari daitzen zaio. Bluetooth-a konektatzea eta informaioa bidaltzea bi exekuzio hari ezberdinetan daude, honen arrazoia, batek konexioa gauzatzen ez badu, besteak informazioa bidaltzea ez saiatzea da.

Bluetooth ziko bat garatu da, honen lana, bateriaren mantentzen du buruan. Bluetooth-a piztea, informazio bidaltzea eta amatzea lortu da notifikazio bat jasotzean. Beraz lehen aipatu den bezala bateria aurrezten da, amaierako erabiltzailearen esperientzia hobetuz

Frogatu egin da konexioa egitean, informazioa zuzen bidaltzen dela, hasieran aipatutako Arduino Serial Monitor tresna erabiliz. Horrez gain Notifikazioak ondo jasotzen eta bidaltzen direla egiaztatzeko, funtzio bat implementatu da. Funtzioak notifikazio bat sortzen du aplikazioaren izenean, honen garapenak beste aplikazioen beharra probak egiteko, notifikazioen sorkuntza inguruan, ezabatu du.

6.2 Pantaila atala

6.2.1 Hardware Muntaiak II

Prototipoari pantaila gehitzeko arazo bat agertu da. Pantailak leku gehiegi hartzen du eta ez da protoboard berdinean sartzen. Beste protoboard bat erabili behar izan da, kableak aldatu behar izan dira, luzeera nahikoa ez zutelako.

Pantailaren muntaiaren barnean Bluetootharen arazo berdina dago. Pantailak 3,3V-tara lan egiten du. Honi aurre egiteko ez da aurreko kasuko bihurtzaile logikoa erabili, HCF4050B-a (ikus. 28) erabili da. Pantailak informazioa bidaltzen ez duelako, ondorioz,

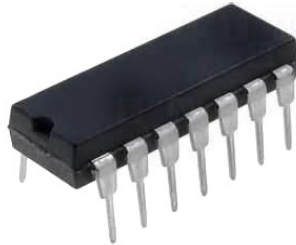


Figure 28: HCF4050B Bihurgailua

informazioa norabide batean bidaltzen denez, norabide bakarreko bihurgailu hau nahikoa da.

Pantaila piztu egiten dela balioztatu da, eta bere funtzionamendua zuzena dela egiaztatzeko grafiko proba bat egin da. Atal honetan u8glib liburutegia erabiliz pantailak zer ikusarazten duen kudeatu da. U8glib liburutegia[1] erabili da, bere konpatibilitatea gure pantailarekin bat datorrelako eta hainbat tresna dituelako gure beharretara egokitzeko. Lehen aipatutako demoa, liburutegi honek dakarren grafiko proba bat da, non irudiak hitzak eta animazioak ikus daitezkeen. Pantailaren demoa kargatu da Arduinoan, honi esker pantailaren norainoko erabilgarritasuna ikusten da.

6.2.2 Arduino Programa II

Demoa alde batera utzita, mezuak pantailaratzeko atala garatu da. Liburutegian datozten ereduak erabilia, pantailaren programazio teknikak ikasi dira, hau garatzeko ahalmena lortuz. Bai letra traimaina bai kokapena, erraz ikusteko eta ulertzeko aukeratu dira.

Lehen adierazitako liburutegia erabilia, datuak non pantailaratuko diren aukeratu da. Ikusi behar den lehenengo datua, Aplikazioaren izena da. Horregaitik hau da gune garrantzitsuena, erdialdea, esleitua duen datua. Letra tamaina, beste datuena baino handiagoa da, arrazoi berdinarengatik, garrantzia. Izenaren aztpian, notifikazioaren texua dago. Egitura honek erraz ulertarazten

dio erabiltzaileari nork zer bidali duen. Goialdeko atala, bi zati ditu, ezkerrean notifikazio kantitatea, eta eskumaldean, Android sistemen antzera, bateria duena(ikus. 29)



Figure 29: Pantailaren egitura

Pantailak informazioa erakutsi dezan, aurretiaz esleitu zaio lekua datu bakoitzari. Hau da, nahiz eta aplikazioak informazio ez jaso, datuei lekua eman zaie. Honela programatu da, pantailaren konfigurazioa aldatu ahal izateko era erraz batean. Beraz nahiz eta guneak guztiz aldatu, informazio irakurketaren atala ez da aldatu behar.

6.3 Konexio Kudeaketa atala

6.3.1 Android Programa III

Propotipoa balioztatu da, baina hainbat atalen hobekuntza garatu behar dela agerian gelditu da. Lehenik, informazioaren bidalketak arazoak sortarazten ditu, hainbat notifikazio batera jasotzen badira. Bigarren, Bluetooth zikloa ez da guztiz osatzen, hau da, ez da bigarren konexio bat jarraian sortaraztea lortu. Hirugarren, informazio eguneraketa dugu, mezurik jaso ezean, informazioa ez da eguneratzen.

Lehenengo arazoari konponbidea aurkitu zaio, Android-eko alarma zerbitzua. Alarma zerbitzua, beste hari batean tenporizadore bat izatearen aurretik aukeratu da. Alarmak gailua pizteko ahalmeda izan ahal duelako, eta bigarren planoan lan egin ahal duelako. Gainera frekuentzia aldatzeko eta piztu-itzaltzeko erraztasunak dakartza.

Alarma kudeatzaile zerbitzua klase berri batean garatu da, eta honi hartzaile baimena esleitu zaio. Alarmaren funtzionamendua garatu da, honek denbora tarte batero dei bat jasoko du. Alarmak bluetooth seinalea bidaltzea implementatu da, lekuz aldatu da deia, notifikazio zerbitzutik alarmara.

Android sistemako pantaila nagusitik, alarmaren kudeaketarako tresnak garatu dira. Hiru botoi gehitu dira, lehenak zerbitzua pizten du, bigarrenak zerbitzua itzaltzen du, eta hirugarrenak frekuentzia aldatzen du. Frekuentzia aldatu eta bost segundura informazioa bidaltzeko implementatu da. Frekuentziak minutuetan kontatzen dira eta gutxienekoa minutu bat da.

Alarma kudeatzailearen implementazioan zehar beste arazo bar dago. Notifikazioen informazioa bidaltzea. Konponbidea, Shared-preferences Android tresna. Shared preferences-en abantaila beste gordetze mekanismoen artean, erraztasuna eta datuen integritatea dira. Erraztasunari buruz bi lerro bakarrik behar ditu datuak gordetzeko, eta beste bi irakurri egiteko. Integritatearen abantaila da, informazioa gordeta gelditzen dela nahiz eta aplikazioa edo sistema huts egin. Beste era batean esanda, behin datu bat lortua, nahiz eta gero aplikazioak huts egin eta mezurik ez jaso, prototipoaren pantaila eguneratu egingo da datu horrekin.

Sharedpreferences tresnarekin, datuak notifikazio zerbitzuan gordetzea eta alarma kudeatzailean irakurtzea lortu da.

Bigarren arazoa, socket edo Bluetooth konexioa trabatu egiten dela da. Aurre egiteko Android-eko tresnak erabiltzea saiatu da, baina ez da konponbiderik aurkitu. Beraz Android developers-en kanpoko irtenbidea erabili da, BluetoothSPP liburutegia.

Liburutegi honek abantaila nabarmenak ekarri ditu. Garrantzitsuena aurreko bi klaseak lerro gutxi batzuegatik ordeztu dituela. Horrek kodearen irakurketa eta mantenimenduan erraztasuna dakar. Gainera log-etako informazioan ikus daiteke Bluetooth konexioaren egoera, arazo bilaketa erraztuz. Erabileraren aldetik hobekuntza nabarmena gauzatu da, Bluetootharen inguruan. Eta Bluetooth zikloaren arazo konpondu egin da.

Notifikazio zerbitzuari dagokionez, azkenengo notifikazioa gordetzen da. Hauen kontaketa ere gorde egiten da, lehenago aipatutako sharedpreferences-etan. Alarmarekin konpondu egin da bi notifikazio batera heltzearen arazoa. Notifikazio bat baino gehiago jasotzen badira, azkenengoa izango da gordeko dena, eta alarmaren txanda heltzean bidaliko da datua. Ondorioz Bluetooth-ari ezingo zaio deitu birritan une berean.

Azkenengo eguneratze arazoaren konponbidea alarma da ere. Honek informazioa denboraro bidaltzen duenez, nahiz eta mezurik ez jaso, bateria eta erlojua eguneratuko dira.

6.3.2 Hardware Muntaia III

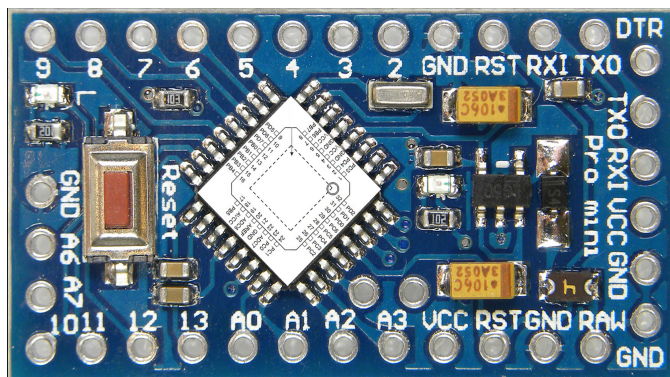


Figure 30: Arduino Pro Mini

Behin egiztatuta softwarea Hardware azken prototipoa sortzen da. Lehenengo konexioen diseinua egin da. Diseinu horren prozesaketa zentroa Arduino Pro Mini-a da. (ikus. 30)

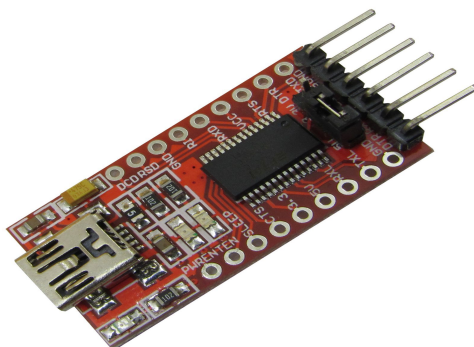


Figure 31: Arduino - Mikrousb bihurtailua

Ikus daitekeenez MiniArduino-ak ez dauka zuzenean ordenagailu bati konektatzeko ahalmena. Horrek arazo larria supozatzen du. Konexiorik gabe ezin dugu programa kargatu, beraz erabilitezina izango genuke. Horretarako arduino pro mini-mikro usb bihurtailua behar dugu.(ikus. 31)



Figure 32: HC6 Bluetooth

Pantaila eta Bluetooth konponenteak ere aldatu dira. Bluetooth-aren kasuan, lehengo RN-44 bluetooth-a, HC6 Bluetooth-ak (ikus.33) ordeztu du. Bluetooth honek Tamaina txikiagoa du eta pin kantidadeari dagokionez, beharrezkoak bakarrik daude.

DOGM128 Pantaila probak egiteko dela argi dago, bere tamaina handiegia da, ezin da eskumuturreko tamainako prototipo batean sartu. Pantaila berriak (ikus.33) 4 pin bakarrik ditu, tamaina txikia eta DOGM128-arentzako erabiltzen den liburutegia erabil dezake.

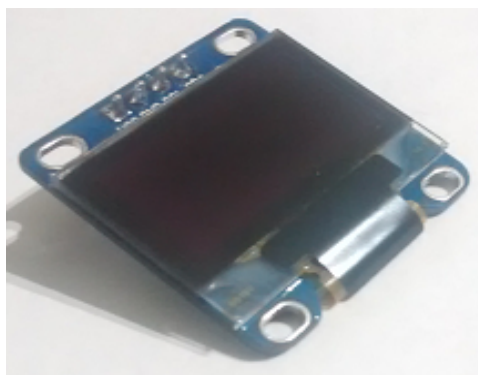


Figure 33: Mini Pantaila

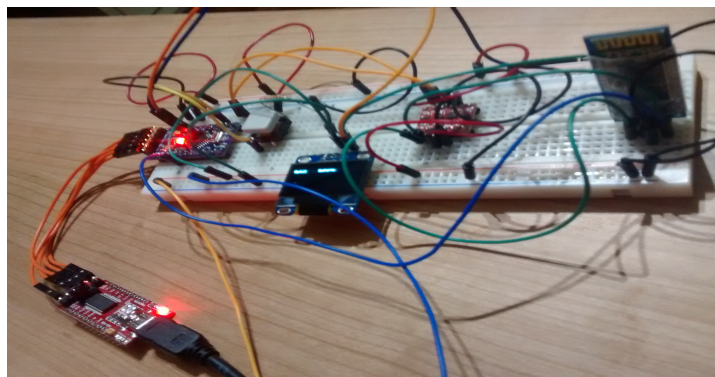
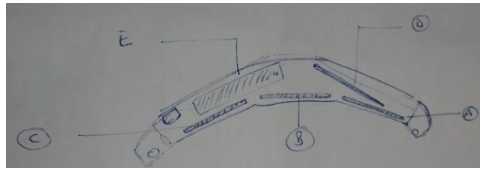


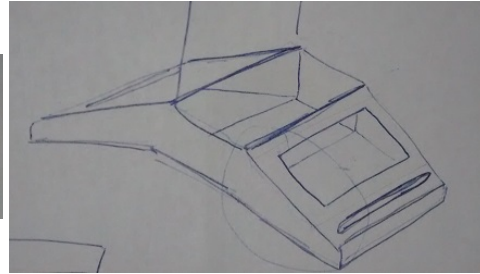
Figure 34: Azken protoboard Prototipoa

Prototipoaren protoboard muntaia egin da (ikus. 34), hardware konponente berriak balioztatzeko.

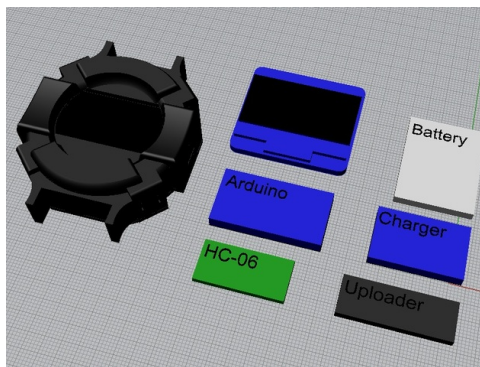
Prototipo honen eskumuturreko muntaia egiteko 3D imprimaketa diseinua garatu egin da.(ikus. 35a, 35b). Lehenengo irudian konponenteen banaketa dugu, hots:



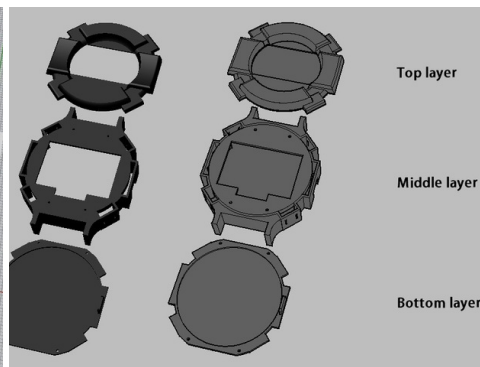
(a) Eskumutur diseinua I



(b) Eskumutur diseinua II



(a) Eskumutur modeloa I



(b) Eskumutur modeloa II

(a) Bluetooth. (b) Arduino Pro Mini. (c) Transformadore. (d) Pantaila. (e) Bateria.

Nahiz eta egindako diseinuak prototipoaren beharrak asetu, Smart-Watch eskumuturren azterketa eginez jada eginda dagoen 3D modeloa lortu da. (ikus. 36a, 36b)[11]. Eskumutur honek, prototipoaren beharrak asetzuz gain, erloju itxura antzekotasun handiagoa dakar.

7 Egiaztapena eta Ebaluzioa

Hurrengo textuan, lanaren atalen probak Kasuak isladatuko dira. Kasue hauek erabiltzaileak gauzatu aha dituen kausen probak dira.

Proba botoia: Erabiltzaileak aplikazioa zabaldu eta proba botoia sakatzen du.

- Esperotako emaitza: Notifikazio bat zortzen da "Botoi Proba" izenburuarekin eta "Mensaje 1" Mezuarekin.
- Lortutako emaitza: Gal notifikazioa sortu egiten da izenburu eta mezu zuzenekin.

Notifikatu Gailu gabe: Erabiltzaileak Bluetooth gailu bat gabe Notifikatu botoia sakatzen du.

- Esperotako emaitza: Mezu bat pantailaratzea, bluetooth gailu bat aukeratu behar duela esanez.
- Lortutako emaitza: Toas bat erakusten du, erabiltzaileak bluetooth gailu bat aukeratu behar duela esanez.

Gailua Aukeratu sakatu BT off: Bluetooth-a itzalita egonda erabiltzaileak gailua aukeratu sakatzea.

- Esperotako emaitza: Mezu bat pantailaratzea, bluetooth a aktibatzeke aukera emanez. Aktibatuta bakarrik igaroko da hurrengo leihora.
- Lortutako emaitza: Mezua zuzen pantailaratzen da. Erabiltzaileak bluetooth-a aktibatu arte ez da hurrengo leihora igaroko.

Gailua Aukeratu sakatu BT on: Bluetooth-a piztuta dagoelarik erabiltzaileak gailua aukeratu botoia sakatzea.

- Esperotako emaitza: Aurreko mezua ez pantailaratzea eta hurrengo leihora igarotzea.
- Lortutako emaitza: Pantaila berria erakusten da mezu gabe.

Gailu berria gehitu: Gailua aukeratu botoia sakatu ostean erabilzaileak gehitu berria botoia sakatzea.

- Esperotako emaitza: Android sistema-ko bluetooth sinkronizaziorako leihoa pantailaratzea.
- Lortutako emaitza: Android sistema-ren pantaila zuzen aurkezten da.

Gailua aukeratu amaitu gabe: Erabiltzaileak gailua aukeratu sakatzea. Sakatu ostean, gailu bat ez aukeratzea, atzera itzultzea eta Notifikatu botoia sakatzea.

- Esperotako emaitza: Mezu bat pantailaratzea, bluetooth gailu bat aukeratu behar duela esanez.
- Lortutako emaitza: Bigarren kasuan bezala toas bat erakusten du, erabiltzaileak bluetooth gailu bat aukeratu behar duela esanez.

Ez notifikatu Zerbitzua Off: Zerbitzua itzalita egonda ez notifikatu botoia sakatzea.

- Esperotako emaitza: Mezu bat pantailaratzea esanez zerbitzua off dagoela.
- Lortutako emaitza: Toast baten bidez erabiltzaileari jakinarazten zaio zerbitzua itzalita dagoela jada.

Frekuentzia aldatu datu gabe: Frekuentzia jarri gabe aldatu frekuentzia botoia sakatzea.

- Esperotako emaitza: Mezu bat pantailaratzea informazioa sartzea eskatuz.
- Lortutako emaitza: Mezua pantailaratzen da.

Frekuentzia aldatu datu okerrekin: Berez zenbaki positiboak bakarrik sartu ahal dira. baina minimoa 1 da, beraz proba hau erabiltzaileak 0 sartzean datza.

- Esperotako emaitza: Mezu bat pantailaratzea informazio zuzena sartzea eskatuz.
- Lortutako emaitza: Mezua pantailaratzen da.

Frekuentzia aldatu zerbitzua off: Zerbitzua piztu gabe, datuekin, frekuentzia aldatu botoia sakatzea.

- Esperotako emaitza: Mezu bat pantailaratzea erabiltzaileari zerbitzua itzalita dagoela jakinaraziz.
- Lortutako emaitza: Mezua zuzen pantailaratzen da.

Zerbitzua itzali: Erabiltzaileak Ez-notifikatu botoia sakatzen du.

- Esperotako emaitza: Zerbitzuak mezu gehiagorik ez du bidaltzen.
- Lortutako emaitza: Mezua zuzen pantailaratzen da.

Frekuentzia aldatu: Erabiltzaileak datuak sartuz Frekuentzia aldatu botoia sakatzen du.

- Esperotako emaitza: Zerbitzuak frekuentzia aldatzen du. Jakinarazteko erabiltzailea, mezu bat pantailaratzen da, zenbateko frekuentziara aldatu den jarriz.
- Lortutako emaitza: Mezua zuzen pantailaratzen da. Zerbitzuaren frekuentzia zuzen aldatzen da.

8 Ondorioak eta etorkizuneko lana

Lan honekin SmartWatch baten oinarrizko funtzionalitateak aztertu dira. Konponente ezberdinekin prototipo bat sortu egin da, Android sistema batekin sinkronizatzeke ahalmena duena.

Bestalde, ikuspegi akademiko batetik, GAL honek karrerako eza-gupen ezberdinak praktikan jartzeko aukera ezin hobea eman dit, aplikazio mugikorrek eta DIY (Do-It-Yourself) elektronika oinarri izan direlarik. Horrez gain, Internet of Things arloan erabiltzen diren elektronikako ezagutzak lortzea ere ahalbideratu dit. Azkenik, produktu baten garapen oso edo integral bat egitea (softwarea eta hardwarea) oso suspergarria izan da.

Kudeaketaren ikuspuntutik, graduan zehar ikusitako hainbat kompetentzia garatu dira. Zentzu honetan, garapenean zehar, proiektu bat modularra izatearen garrantziaz ohartu egin naiz. Honi esker proiektuan zehar sortutako arazoan aurrean, erraztasunez egin dira konponketak. Proiektua atalka egiteak, kudeaketa erraztasunak ekarri ditu. Helburu txikiak jarritz, eta emaitza edo garapen bat aurrekoaren gainean oinarrituta, pixkanaka proiektuak forma hartu izan du.

Hasierako proposamenetik ez da aldaketa askorik egin. Aldaketak diseinuan egin dira, bai diseinuaren hobekuntza posiblea zelako markatutako denbora, edo aurreikusitako emaitzak lortzen ez zirelako. Hauen artean Bluetooth-aren kasuda dugu nabariena. Hasierako diseinua baztertu behar izan zen, trenek ez zutelako gure behar guztiak asetzen eta horregaitik arazoak zekartzaten.

Etorkizunari begira prototipo honek haibat hobeguntza izan ditzake.

Gps funtzionalitatea: bai mugikorreko GPS-a erabiliz edo barneko konponente baten datuak irakurriz informazioa bidaltzea.

Notifikazio aukeraketa: Aplikazioen konfigurazio txosten bat sortzea, eta aplikazio bakoitzeko zein datu bidali nahi den eta zein ez konfiguratu ahal izatea.

Dei alarmak: Dei bat heltzerakoan eskumuturrekoak, bai soinu, argiz edo bibrazioz jakinaraztea erabiltzailea.

Leiho konfigurazioa: Erabiltzailearentzako tresnak, prototipoaren pantaila konfiguratu ahal izateko.

Prototipo Erantzunak: prototipotik erantzunak bidali ahal izatea android sistemari.

Hobekuntza hauek hainbat ordu eramango lituzke pertsona bakar batentzat, baina modularitateari eta hizkuntza ulergarriari esker, kodea erraz uler daiteke. Kontuan hartu behar da, prototipo honek loop baten bidez lan egiten duela. Hau da, ez su Sistema Eragilerik. Beraz bere mugak ditu.

Dokumentuan isladatu den bezala, lana aurrera zioala arazoak agertzen joan dira. Arazo hauetatik hainbat gauza ikasi ditut. Gehienbat, nahiz eta dena ondo antolatua izan, beti hobetu egin ahal dela zeozer, eta gauzak txarto ateratzea normala dela. Gauzak inoiz ez dutela lan egiten lehenengo aldiz, eta zuk izan duzun arazoa beste norbaitek konpundu izan ahal duela. Pertsona horrek lagundu bazaitu zure bidean ziurrenik zuk konpondutako beste atal bat, biharko egunean norbaiti lagungarria izango zaio. Hori kontuan haru dut kodea librea erabakia egin beharreko momentuan. Gainera horrela, zure proiektuaren gainean beste norbaitek idea interesgarri bat izan dezake.

9 Bibliografia

References

- [1] U8glib library. <https://code.google.com/p/u8glib/>.
- [2] akexorcist. Android-bluetoothspplibrary, 2014. <https://github.com/akexorcist/Android-BluetoothSPPLibrary>.
- [3] Mikal Hart. Softwareserial library. <https://www.arduino.cc/en/Reference/SoftwareSerial>.
- [4] BI Intelligence. The wearables report: Growth trends, consumer attitudes, and why smartwatches will dominate. *Business Insider*, 2015. <http://www.businessinsider.com/the-wearable-computing-market-report-2014-10>.
- [5] Nicole Lee. Moto 360 review: It's the best android wear watch, but that isn't saying much. *engadget*, sep 2014. <http://www.engadget.com/2014/09/12/moto-360-review/>.
- [6] Motorola. Developer resources, 2015. <http://www.motorola.com/us/Motorola-Developers/motorola-developers.html>.
- [7] qwerty. Bluetooth terminal, 2015. <https://play.google.com/store/apps/details?id=Qwerty.BluetoothTerminal>.
- [8] Will Shanklin. 2014 smartwatch comparison guide. *gizmag*, nov 2014. <http://www.gizmag.com/compare-best-smartwatches-2014/34880/>.
- [9] smartwatchgroup. Overview of the smartwatch industry. 2014. <http://www.smartwatchgroup.com/overview-smartwatch-industry/>.
- [10] smartwatchgroup. Top 10 smartwatch companies 2014 (sales). 2014. <http://www.smartwatchgroup.com/top-10-smartwatch-companies-sales-2014/>.
- [11] Wonho. Arduino smart watch, 2014. <http://www.thingiverse.com/thing:610603>.