



**BILBOKO INDUSTRIA INGENIARITZA TEKNIKOKO
UNIBERTSITATE ESKOLA**



INDUSTRIA ELEKTRONIKAREN ETA AUTOMATIKAREN INGENIARITZAKO GRADUA

GRADU AMAIERAKO LANA

2016 / 2017

*HARDWARE IREKIAREN ETA GAUZEN INTERNETAREN APLIKAZIO BAT:
"HODEIRA" KONEKTATUTAKO EGURALDI BEHATOKIA*

GRADU AMAIERAKO LANA

IKASLEAREN DATUAK

IZENA: BORJA

ABIZENAK: CASTRILLO LEON

SIN.:

DATA: 2017/02/10

ZUZENDARIAREN DATUAK

IZENA: OSKAR

ABIZENAK: CASQUERO OYARZABAL

Saila: Sistemen Ingenieritza eta Automatika

SIN.:

DATA: 2017/02/10



AURKIBIDEA

1. Sarrera.....	1
2. Helburuak	4
3. Deskribapena	5
4. Alternatibak	7
4.1. MIKROKONTROLAGAILU PLAKAK.	7
4.1.1. Arduino.....	7
4.1.2. Rasberry PI	10
4.1.3. Intel Edison.....	12
4.2. Haririk gabeko komunikazioa.....	13
4.2.1. RF433 Mhz	13
4.2.2. XBee	14
4.2.3. Bluetooth	15
4.2.4. WiFi.....	16
4.3. Kanpo behatokiarentzako elikadura	16
4.3.1. Nikel-Metal BATERIA	17
4.3.2. Alkalina bateriak.....	17
4.3.3. Saretik.....	18
4.3.4. LiPo bateriak.....	18
5. Aukeratzeko irizpideak.....	19
5.1. Prezioa	19
5.2. Kontsumoa.....	19
5.3. Tentsio nominala	19
5.4. Kargatzeko ahalmena	19
5.5. Komunikazio tartea.....	20
5.6. Funtzionamendua.....	20
5.7. Erabilgarritasuna.....	20
5.8. Tamaina	20
5.9. Hautaketa taulak	21
6. Konponenteak.....	23
6.1. Sentsoreak.....	23
6.1.1. Barneko sentsorea.....	23
6.1.2. Kanpoko sentsorea.....	25



6.2. XBee-ak.....	26
6.2.1. XBee Adapter Board eta USB portuaren arteko konexioa	27
6.2.2. Kanpo behatokiko xbee Adapter Board eta USB portuaren arteko konexioa.....	28
6.2.3. Barne behatokiko xbee Adapter Board eta USB portuaren arteko konexioa.....	29
6.2.4. XBee-ak konfiguratze eta egiaztatze programa	31
6.3. Pantaila	33
7. Behatokiak	38
7.1. Kanpoko behatokia.....	38
7.1.1. Kanpo behatokiko konexioak	38
7.1.2. Kanpo behatokiko programazioa	40
7.2. Barneko behatokia	42
7.2.1. Barne behatokiko konexioak	43
7.2.2. Barne behatokiko programazioa	44
8. Energia kontsumoa	46
8.1. Arduino promini-aren energia kontsumoa.....	46
8.1.1. Arduino promini lokartzeko programazioa	47
8.2. XBee-aren energia kontsumoa.....	51
8.2.1. XBee-aren lokartzeko programazioa	53
9. Datuak hodeira igo.....	59
10. Software programak	65
10.1. Arduino programa.....	65
10.2. X-CTU programa.....	66
10.3. Fritzing programa	66
10.4. DesignSpark programa	67
11. Etorkizunerako hobekuntzak	;Error! Marcador no definido.
12. Bibliografia.....	72



IRUDIEN AURKIBIDEA

1.irudia: Kartel elektronikoa.....	2
2.irudia: Bloke diagrama	6
3.irudia: Freduino UNO.....	8
4.irudia: Arduino ProMini.	9
5.irudia: Arduino YunMini.	9
6.irudia: Arduino Nano.	10
7.irudia: Raspberry Pi A+.	11
8.irudia: Raspberry Pi B+.	11
9.irudia: Raspberry Pi 2B.	12
10.irudia: Intel Edison.....	13
11.irudia: RF433 igorlea.	14
12.irudia: RF433 hartzailea.	14
13.irudia: XBee.	15
14.irudia: Bluetooth HC-06.....	15
15.irudia: WiFi Serial ESP8266.....	16
16.irudia: Nikel-Metal bateriak.	17
17.irudia: Alkalina bateriak.	17
18.irudia: Sare kablea.....	18
19.irudia: 3,7 V-eko LiPo bateria.	18
20.irudia: Arduino eta sentsore konexioa.....	23
21.irudia:DHT22 pin-etako argazkia eta arduinoarekin duen konexioa irudia.	25
22.irudia: XBee adapter pin-ak.....	27
23.irudia: FTDI eta XBee konexioa.....	28
24.irudia: FT232R-aren pin-ak eta pin-en deskribapena.	29
25.irudia: FT232R, J3 konexio pin-ak.	29
26.irudia: FT232R eta XBee konexioa.	30
27.irudia: Barne behatokiko XBee-aren datuak.	31
28.irudia: Kanpo behatokiko XBee-aren datuak.....	32
29.irudia: Maila bihurgailu eta pantaila pin-ak.	33
30.irudia: Barne behatokiko arduino, pantaila eta maila bihurgailu konexioak.....	35
31.irudia: Pantailaren datasheet datuak.	35
32.irudia: Arduino programa.....	36
33.irudia: Arduino programa.....	37



34.irudia: Kanpo behatokiko konexioak.....	39
35.irudia: Programa arduinora igotzeko konexioa.....	40
36.irudia: Kanpo behatokiko arduino programa.....	42
37.irudia: Barne behatokiko arduinoa, XBee-a eta sentsorearen arteko konexioa.....	43
38.irudia: Barne behatokiko arduino programa.....	45
39.irudia: Arduino ProMini-ren BEREZKO korrante kontsumoa mA-tan.....	46
40.irudia: Arduino ProMini datasheet (54. orrialdea, 11.9.1. atala).....	47
41.irudia: Arduino ProMini programa.....	47
42.irudia: Arduino ProMini programa.....	47
43.irudia: Arduino ProMini datasheet(54. orrialdea, 11.9.2. atala).....	48
44.irudia: Arduino ProMini datasheet (55. orrialdea, 11.9.2. atala).....	48
45.irudia: Arduino ProMini programa.....	49
46.irudia: Arduino ProMini datasheets (55. orrialdea, 11.9.2. atala).....	49
47.irudia: Kanpo behatokiko arduino programa.....	50
48.irudia: Arduino ProMini-aren korrante kontsumoa LO dagoenean (POWER_DOWN lo modua).....	50
49.irudia: XBee-aren BEREZKO korrante kontsumoa mA-tan.....	53
50.irudia: XBee-aren datasheet (25. orrialdea, "command mode" atalean).....	54
51.irudia: XBee-aren datasheet (23. orrialdea, "Sleep mode" atalean).....	54
52.irudia: XBee-aren arduino programa.....	56
53.irudia: XBee-aren korrante kontsumoa mA-tan, LO dagoenean (Hibernate lo modua).....	56
54.irudia:Arduinoa eta XBee-aren arteko BEREZKO korrante kontsumoa mA-tan.....	57
55.irudia:Arduinoa eta XBee-aren arteko korrante kontsumoa mA-tan, LO dagoenean.....	58
56.irudia: ThingSpeak orria.....	59
57.irudia: ThingSpeak web-guneko datuak.....	60
58.irudia: Aginduak idazteko modua.....	60
59.irudia:Arduino programan programatutakoa.....	62
60.irudia: ThingSpeak-en behatokietako datuak marraztu grafikoan.....	63
61.irudia: Matlab-en datuak aurkeztuak.....	64
62.irudia: Arduino programa.....	65
63.irudia: X-CTU programa.....	66
64.irudia: Fritzing programa.....	67
65.irudia: DesignSpark programako konponente konexioak.....	68
66.irudia: DesignSpark programan konexio finala.....	68
67.irudia: Zirkuitua azetato papelean inprimatua.....	69
68.irudia: Kobrezko plaka eta azetato papela, insoladore makinan.....	69
69.irudia: Kobrezko plaka atakatzea.....	70



<i>70.irudia: Kobrezko plaka garbitu ondoren.....</i>	<i>70</i>
<i>71.irudia: Kobrezko plakaren azken emaitza.</i>	<i>71</i>



TAULEN AURKIBIDEA

<i>Taula 1: Hautaketa irizpideak</i>	21
<i>Taula 2: Plaka mikrokontroladore hautaketa</i>	21
<i>Taula 3: Haririk gabeko komunikazio hautaketa</i>	21
<i>Taula 4: behatokiko elikadura hautaketa</i>	22
<i>Taula 5: Arduino eta sentsore konexio deskribapena</i>	24
<i>Taula 6: Arduino eta sentsore konexio deskribapena</i>	26
<i>Taula 7: USB eta XBee adapter konexioa</i>	28
<i>Taula 8: USB eta XBee adaptadore konexio deskribapena</i>	30
<i>Taula 9: Arduino, maila bihurtu eta pantaila konexio deskribapena</i>	34
<i>Taula 10: Arduino ProMini eta XBee adaptadore konexio deskribapena</i>	39
<i>Taula 11: Arduino ProMini eta bateria konexio deskribapena</i>	39
<i>Taula 12: Arduino ProMini eta USB konexio deskribapena</i>	40
<i>Taula 13: Arduino YunMini eta XBee adaptadore konexio deskribapena</i>	43



1. SARRERA

Datuz aberatsa den garai batean bizi gara. Hasiera batean, datu aberastasun hori plataforma digitaletan (Web 2.0 tresnak, sare sozialak) pertsonak gordetako datuetatik eratorri da. Baina azken aldi honetan, "Gauzen Internetaren" etorreratik hona, alegia, mundu erreal edo fisikoko magnitude fisikoak neurtzeko eta lagin horiek hodeian gordetzeko aukera azaldu da, hala nola: klimarekin erlazionatutako datuak (tenperatura, hezetazuna, olatuen altuera, gas ezbedinen kutsadura maila, etabar), ikusmenarekin erlazionatutako datuak (manifestaldi batean dagoen pertsona kopurua, bidegurutze bateko semaforo batean itxoiten dagoen kotxe kopurua, etabar), industria jakin baten funtzionamenduarekin erlazionatuako datuak (adibidez, botilen fabrikazioan: fabrikatutako botila kopurua, botilen tenperatura labetik irtetzerakoan, etabar). Behin hodeian (publiko edo pribatuan) daudelarik, datuak algoritmo ezberdinak aplikatuz aztertu eta bistaratu daitezke, erabiltzailearentzako balio gehigarria emanez.

Gradu amaierako lan hau testuinguru honetan kokatu behar da. Hain zuzen ere, lan honek "gauzen internetaren" plataforma txiki bat garatzea helburu du, mundu errealeko magnitude fisikoak sentzore sareen bitartez zelan eskuratu, hodeira igo eta aztertu daitezkeen eredu adieraziz. Eredu adierazpen hau gauzatzeko, eguraldi behatoki baten adibidea hartu da ardatz bezela, adibide honek, alde batetik, aurreko puntuan adierazitako alderdi guztiak garatzeko aukera aurkezten baitu, eta, bestetik, honelako lan berri batean aurkezten diren ideia berriak argi ikusi daitezkeen beharrezkoak diren sinpletasuna eta zehaztasuna baititu.

Eguraldi behatoki baten funtzioa, eguraldiaren berri ematea da. Eguralditik informazio ezberdinak jaso daitezke adibidez; egiten duen tenperatura, zenbateko presioa egiten duen, zenbateko hezetasuna dagoen, haize abiadura... baina gehien bat tenperaturari buruzko informazioa plasmaturik agertzen da. Gaur egun, informazio hori, komertzio ezberdineko kartel elektronikoetan ikusgarri egoten da. Kartel elektroniko horietan, normalean, orduaz eta tenperaturaz informatzen da.



1.irudia: Kartel elektronikoa

Temperaturaz informatzen duten kartel elektroniko hauek, antzeko konponente elektroniko osatuta daude gehienak. Kartel batzuetan, nahiz eta informazio ezberdinak adierazi, montaje bera dute. Karte elektronikoak, eguraldi behatokiak direla esan daiteke, bertatik tenperatura irakur daitekela. Behatoki hauek izaten duten konponenteak hauek izaten dira:

- Plaka mikrokontroladore bat, non programa osoa burutzen da eta konponenteen elkar konexioak burutzen dituena.
- Sentsore bat, neurtu nahi diren ezaugarriak neurtzen duena.
- Pantaila bat, datuak inprimatzeko eta datu horiek ikusgai izateko.
- Elikadura iturri bat, zirkuitua elikatzeko sare elektrikoaren bidez edo bateria batzuen bidez.

Behatokiak, konponente gehiagoz osaturik egon daitezke baina funtzio simple bat egiteko, goian aipatu diren konponentekin nahikoa izan litzake. Gaur egun, mikrokontrolagailu plaken hardware irekiko aukerarekin, posibilitate ezberdinak irtetzen dira behatokiak kontrolatzeko adibidez, behatokiak neurtu dituen datuak beste toki batera bidaltzea, han ere ikusi ahal izateko edota datuak hodeira igoz.



Gradu Amaierako Lan (GAL) honetan, eguraldi behatokiaren proiektua garatzean, gradu osoan eskuratutako adimena praktikan jarriko da eta gai berriei buruzko informazioarekin adimen berriak jasoz.



2. HELBURUAK

Gradu Amaierako Lan (GAL) honen helburu nagusia, bi behatokien diseinu eta programazioa egitean datza. Helburu nagusi hauek lortzeko, lehenengo azpi helburu batzuk bete behar dira, azkenengo helburu nagusiak beterik izateko. Lan honetan bete beharreko azpi helburuak, honako hauek dira:

- Hezetasun eta tenperatura neurtzen dituen bi behatokien muntaketa.
- Kanpoko eta barneko behatokien arteko konfigurazioa, bien arteko komunikazio izateko.
- Kanpo behatokiko tenperatura eta hezetasuna neurtzea.
- Kanpo behatokiko datuak, barne behatokira bidaltzea.
- Barne behatokiko tenperatura eta hezetasuna neurtzea.
- Barne behatokian, kanpoko behatokiko datuak irakurtzea.
- Bi behatokien datuak, barneko behatokiko pantailan inprimatzea.
- Kanpo behatokia, bateria batez elikatzea.
- Kanpo behatokiko konponente elektronikoei, energia aurrezteko programatzea:
 - Arduino ProMini-a lokartzeko prestatu
 - XBee-a lokartzea
- Pantailan inprimatu diren, kanpoko eta barneko behatokien neurketak, hodeira igotzea eta grafiko batean bidez irudikatzea.
- Datuak eskuragarri izatea, egin nahi diren kalkulu estadistikoak egiteko.

Behatokien eta konponente elektronikoen guztien arteko funtzionamendua eta programazioa egiteko, hardware eta software librean oinarritutako platafoma libreko plaka batekin burutuko da kontrola.



3. DESKRIBAPENA

Bi eguraldi behatoki dira, non batek leku bateko tenperatura eta hezetasuna neurtuko duena eta bestea aldiz, beste leku batetako tenperatura eta hezetasuna neurtuko duena.

Barneko behatokiaren muntaketa egiteko erabiliko diren konponente elektronikoak hauek izango dira; arduino YunMini mikrokontrolagailu bat, irrati frekuentziazko XBee txip bat, LCD pantaila bat eta HTU21D sentsore bat (tenperatura eta hezetasuna neurtuko duena).

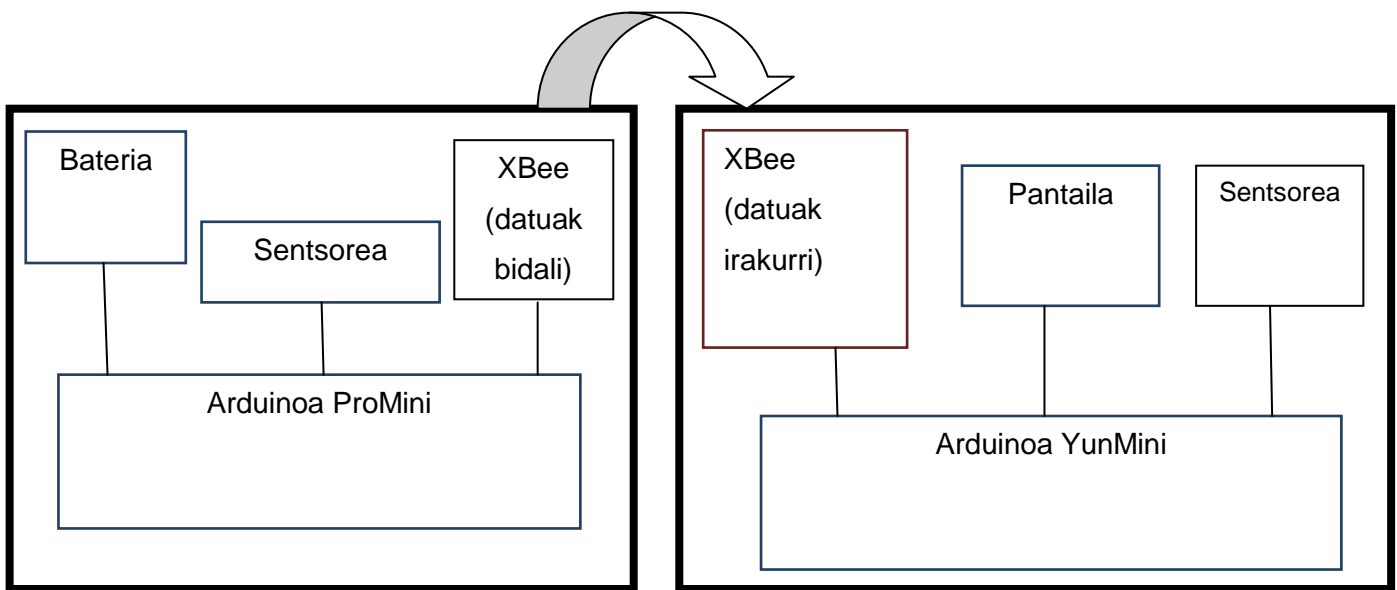
Kanpoko behatokia aldiz, konponente elektroniko gutxiago izango dira; kasu honetan, arduino ProMini-a bat izango da, irrati frekuentziazko beste XBee modulu berdin bat, batería bat eta beste sentsore ezberdin bat. Aukeratu den sentsorea DHT22(AM2302) da, honek ere tenperatura eta hezetasuna neurtzen ditu.

Bi behatoki ezberdin daudela jakinik, euren funtzionamendua honakoa izango da. Barneko behatokiko sentsoreak, tenperatura eta hezetasuna neurtzen egongo dira eta arduino YunMini-aren memoriari pasatuko dizkio datuak. Kanpo behatokiak ,aldiz, DHT22 sentsorearen bidez neurtuko ditu tenperatura eta hezetasuna, eta arduino ProMini-aren memoriara pasatuko dizkio. Kanpo behatokiko arduinoak, XBee-ari pasatuko dizkio datuak. Kanpo behatokiko XBee-ak, barneko behatokiko XBee-ari bidaliko dizkio datuak irrati frekuentziaren bidez. Gero, barne behatokiko XBee-ak arduino YunMini-ari pasatuko dizkio datuak eta honek bere memorian gordeko ditu. Datuak arduino YunMini-an izanda, pantailan inprimatuko dira.

Kanpo behatokia, ez da denbora guztian egongo balioak neurtzen eta bidaltzen, baizik eta, 20 segundurik behin neurtu eta bidaliko ditu. Datuak 20 segunduro bidaliko dira, denbora horretan arduino ProMini-a eta XBee-a lo moduan utziz. Horrela, zirkuituaren potentzia kontsumoa minimizatzea lortzen da.

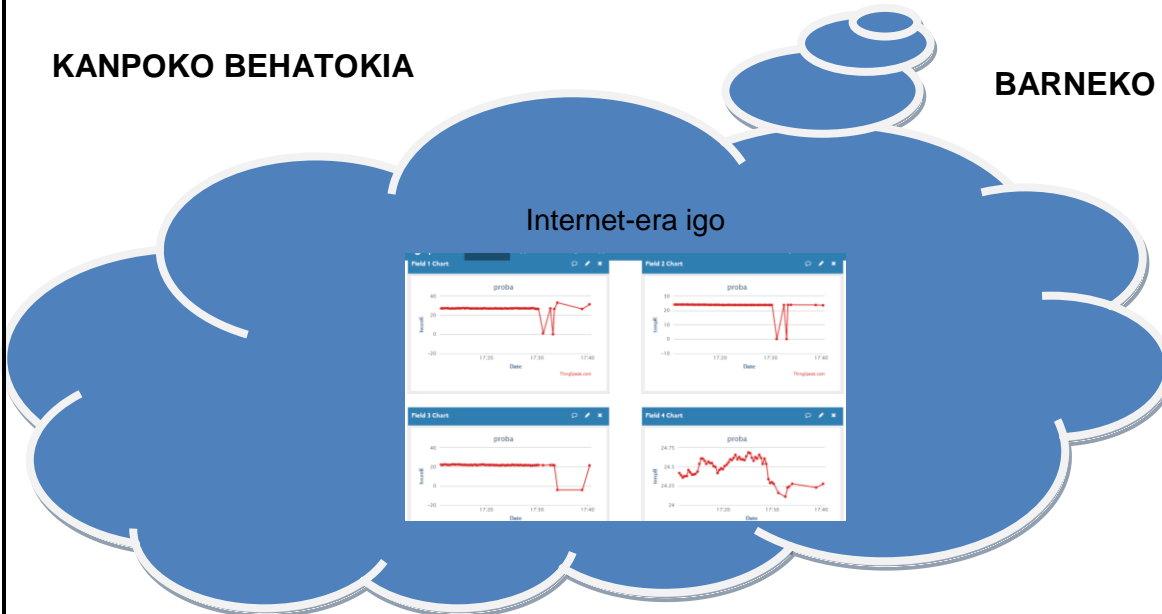


Pantailan inprimatutako datuak, arduino YunMini-ari esker internet-era igoko dira datuak eta grafiko batean erakutsi ahal izango dira. Datu horiek hodeian izanik aparte, Matlab-en bidez ikertzeko aukera izango da, nahi diren algoritmo bitartez kalkulu ezberdinak egiteko.



KANPOKO BEHATOKIA

BARNEKO BEHATOKIA



2.irudia: Bloke diagrama



4. ALTERNATIBAK

4.1. MIKROKONTROLAGAILU PLAKAK.

4.1.1. ARDUINO

Arduinoa, hardware eta software librean oinarritutako plataforma ireki elektronikoa da, proiektu elektroniko ezberdinak diseinatzeko egina. Plataforma hau bi ardatzetan oinarrituta dago:

- **Hardware ardatza.** Mikrokontrolagailua erosotasunez erabiltzeko plaka, hots: mikrokontrolagailua elikatzeko tentsio bihurgailua, mikrokontrolagailua martxan jartzeko kuartzozko kristala, mikrokontrolagailua programatzeko serie seinaleak USB-tik TTL-ra pasatzen dituen bihugailua eta mikrokontrolagailuaren pinak erosotasunez erabiltzeko zokaloak, besteak beste, gehiturik dituena.
- **Software ardatza.** Mikrokontrolagailua exekutatu diren programak programatzeko garapen ingurunea. Garapen ingurunea hiru elementuz osaturik dago: (1) C/C++ antzeko programazio lengoia; (2) programazioa errazten duten oinarritzko hainbat liburutegi; eta (3) programa konpilatze eta mikrokontrolagailura igotzeko baliabideak, adibidez: Atmel-ek zuzenean saltzen dituen mikrokontrolagailuak ez bezala, Arduinok dakarren Atmel mikrokontrolagailua "boot loader" batekin dator.

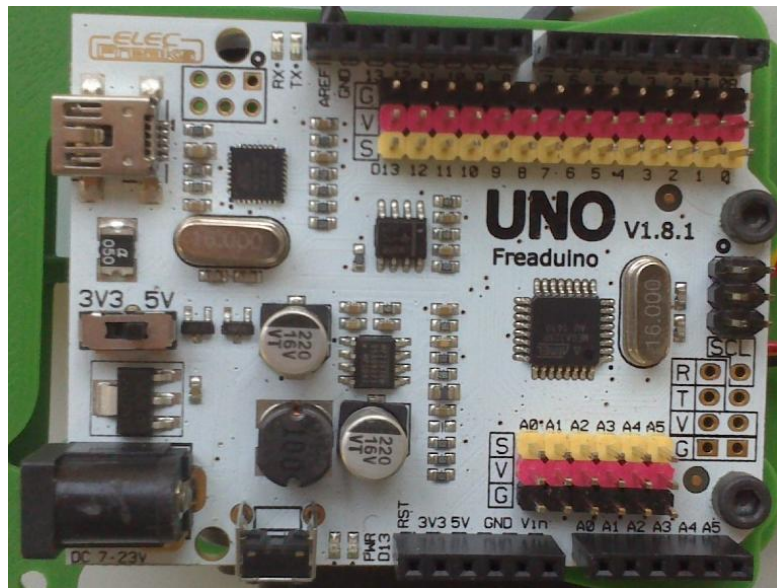
Arduino plaka ezberdinak existitzen dira, aukera ezberdinak honakoak dira:

- **Freaduino UNO:** arduino UNO bezalako plaka elektronikoa dauka, ATmega328 mikrokontrolagailuarekin. 14 sarrera/irteera digital ditu, horietatik 6 PWM irteera bezala erabili daitezkenak eta beste 6 sarrera analogiko. Gainera, 16 Mhz-ko erresonadore zeramikoa du, USB konektore bat, elikadura konektore bat, ICSP goiburu bat eta reset botoi bat. Plakak behar duen guztia darama,



mikrokontrolagailua lan egiteko. Bakarrik falta dena da, USB konexio baten bidez ordenagailura konektatzea edo transformadore bateri konektatzea energia elektrikoa izateko.

- Freaduino UNO honek, IDE arduino programarekin %100 bateragarria da. Arduino honek, Vcc eta GND pin-ak ondo adierazita ditu, lagungarri izateko sentsorea, motorea, edo beste konponente bat konektatzean. Arduino irteera guztiak 3,3 V-eko erreferentzia dute baina mikrokontrolagailua 5 V-etan lan egindo du. Freaduino UNO honek, maila bihurtu bat darama barnean operaketa ezberdinak egiteko eta konexio gehigarriak ez egiteko.



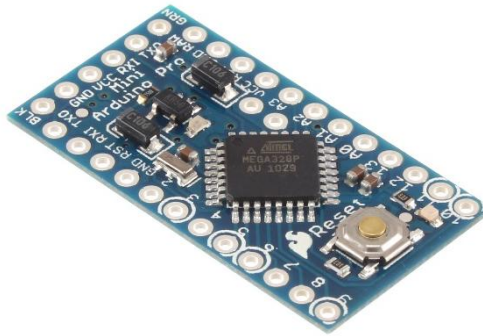
3.irudia: Freaduino UNO.

- Arduino ProMini-a: Arduino ProMini, ATmega328 txartean oinarrituta dago. Hamalau sarrera/irtera pin digital ditu (horietako 6 PWM irteerak dira), 6 sarrera analogiko, erresonadore bat, berrabiarazte botoi bat eta konektoreak jartzeko zuloak. Sei pin-eko konektorea jarri ahal zaio, FTDI kablea edo FTDI txartela (FTDI Basic Breakout) konektatzeko eta USB batekin komunikazioa zirkuituan izateko.

Arduino ProMini-a diseinatuta dago konponentetan instalazio ia iraukorra izateko. Txartela konektorerik gabe dator, horrela erabilera ezberdinak aukeratu daitezke



aplikazio ezberdinentzat edo zuzenki kableak soldatuta. Arduino honek, energia aurrezteko barne programatu daiteke.



4.irudia: Arduino ProMini.

[https://upload.wikimedia.org/wikipedia/commons/0/01/Arduino_Pro_Mini_\(2\).jpg](https://upload.wikimedia.org/wikipedia/commons/0/01/Arduino_Pro_Mini_(2).jpg) -tik aterata

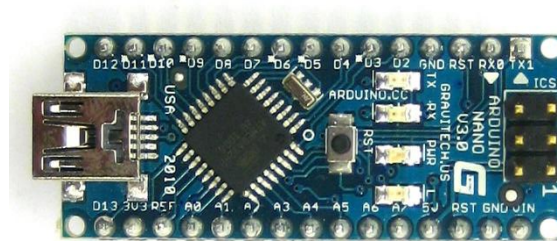
- **Arduino YunMini-a:** ATmega32u4 oinarritutako mikrokontrolagailu bat du eta Linux modulu bat ere. Biak elkarrekin konektatuta daude baina bata bestearekin konektatzeko Bridge liburutegia erabiltzen da. Plaka honen abantaila bat da, Linux moduluaren bidez WiFi-z programatu daitezkeela. Arduino honek, bi sare konexio ditu, bata 10/100 mbps Ethernet sarea eta bestea WiFi. 20 sarrera/irteera digital ditu, horietatik 7 irteera PWM-arentzat eta 12 sarrera analogikoentzat. 16MHz kuartzoko kristala darama, USB mikro konektore portua dauka, ICSP goiburuak, bi reset botoi eta erabiltzaile botoi bat.



5.irudia: Arduino YunMini.



- Arduino Nano: ProMicro honek, ProMini-aren antzerakoa da baina honek, ATmega32U4 mikrokontrolagailu batekin. Txip honek, gizakion interfaze moduan lan egin dezake. USB transzeptorea 32u4 USB konektibitatea du eta kanpoko USB-a. Arduino funtzio guztiak betetzen ditu txartel txiki honek: 10 bit-eko ADC 4 kanal, PWM 5 pin, 12 DIO, hardware konexioa Rx eta Tx seriean. 16MHz eta 5 V-etan funtzionatzen du. Mikrokontroladore honek edozein lekutan jar daiteke. Tentsio erregulatzailerik bat du barnean 12 V arte jasan dezakena.



4.irudia: Arduino Nano.

https://upload.wikimedia.org/wikipedia/commons/8/8d/Arduino_Nano.jpg -tik aterata

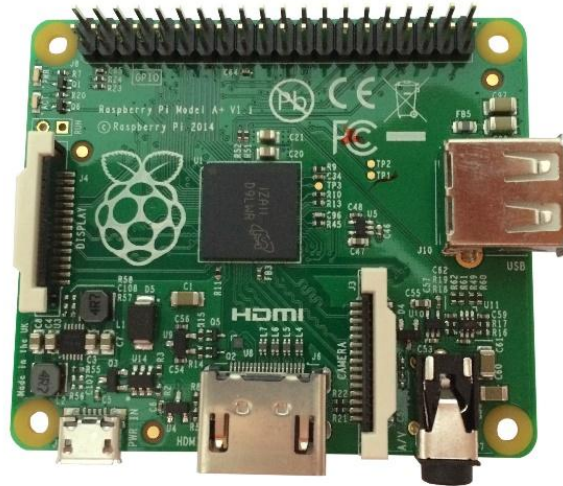
4.1.2. RASBERRY PI

Raspberry Pi da, plaka murriztuko ordenagailua edo (plaka bakarreko) (SBD), koste baxukoa. Diseinua System-on-a-chip Broadcom BCM2835 dakar, prozesadore nagusia dauka (CPU) ARM1176JZF-S 700 MHz-tara, prozesadore grafiko bat (GPU), VideoCore IV eta 512 MB RAM memoria. Diseinua ez dakar disko gogorrik, SD txartel bat dauka biltegitratzeko; ez dauka elikadura iturririk, ez dauka karkasarik.

- A+: Raspberry Pi A+ plakak, mini PC bat integratua dauka, Broadcom BCM2835 SoC Full HD prozesadorea du, GPU Co-procesador multimedia Dual Core VideoCore IV-a du, 256 MB SDRAM eta 700 MHz-ko RAM memoria. microSD txartel



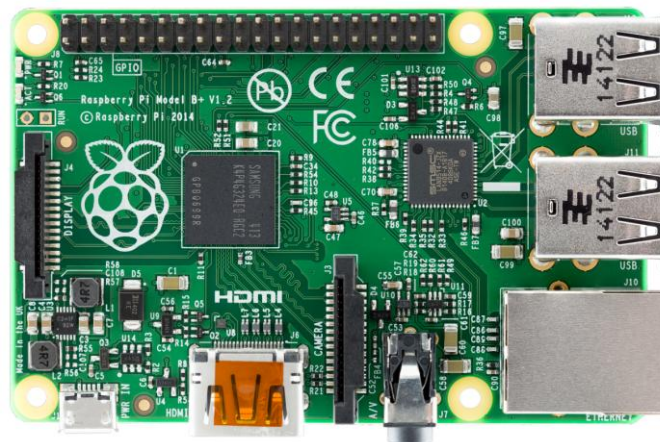
biltegitratzea, USB portu bat eta HDMI irteera, 40 pin-eko GPIO eta 600mA-ko energia kontsumoa.



7.irudia: Raspberry Pi A+.

https://upload.wikimedia.org/wikipedia/commons/3/30/Raspberry_Pi_A%2B.jpg -tik aterata

- B+: Raspberry Pi B+ plaka Broadcom BCM2835 (CPU, GPU eta SDRAM) prosezadorea, 700 MHz-ko CPU ARM1176JZF-S bat, GPU Broadcom VideoCore IV bat, 512 MB-eko RAM memoria (GPU-arekin bateragarria), lau USB 2.0 konexio, 3,5 mm-ko mini jack audio irteera, audio/video HDMI irteera, LAN 10/100 eta WiFi sarrera konektibitatea, mikro SD txartela sartzeko portua bat du ere.



8.irudia: Raspberry Pi B+.

https://upload.wikimedia.org/wikipedia/commons/6/6f/Raspberry_Pi_B%2B_top.jpg -tik aterata



- Raspberry Pi 2 B: Raspberry Pi 2 B plaka ordenagailua, Raspberry Pi Fundatio aktualizazioa duen bakarra da.

Bere ARM A7 prosezatzaileari esker, GNU/LINUX edo Windows 10 banatzea erabil dezake, gainera B+ modeloaren format bera dauka. 900 MHz-ko Quad-Core Cortex A7 da bere prosezadorea, 1 GB-ko memoria Ram-arekin, USB 2.0 lau portuekin, 40 pin GPIO, HDMI irtera bat, Etherne portu bat, audio/mic konbo bat, kamara interfaze bat (CSI), pantaila interfazea (DSI), SD mikro txartela eta 3D nukleo grafikoa.



5.irudia: Raspberry Pi 2B.

[https://upload.wikimedia.org/wikipedia/commons/3/31/Raspberry_Pi_2_Model_B_v1.1_top_new_\(bg_cut_out\).jpg](https://upload.wikimedia.org/wikipedia/commons/3/31/Raspberry_Pi_2_Model_B_v1.1_top_new_(bg_cut_out).jpg)

–tik aterata

4.1.3. INTEL EDISON

Intel Edison zentruan, 2nm-ko SoC Intel Atom-a 500 Mhz-ko nukleo bikoitzeko CPU eta 100 MhZ MCU. Produkua kopila daiteke eta datuak MCU kontsumo baxu bitartez prosezatzen ditu eta CPU analisisia egiteko datuen translazioa filtratuko da. Edison-ek du memoria GB 1, biltegitatze 4 GB, WiFi banda bikoitza eta 4.0 Bluetooth-eko komunikazioa. Konfigurazio aukera ezberdinak ezkeintzen ditu 4 GPIO.



Plataforma berri bat da, erabiltzailearentzat ez ditu datu euskarri oso onik eta periferikoentzako ez du sistema handirik.



6.irudia: Intel Edison.

https://c2.staticflickr.com/8/7493/15314330503_3715218321.jpg -tik aterata

4.2.HARIRIK GABEKO KOMUNIKAZIOA.

4.2.1. RF433 MHZ

RF433 MHz moduluak simplex datu erlazioekin funtzionatzen dute, hau da, sentzu batez informazioa transmititzen dute. Konponente hau, sentzu bateko komunikazioa egiteko erabil ohi da. Edozein mikrokontroladoreari edo kodifikadore/dekodifikadore zirkuituari erraz konektatu daiteke, RF erlazio arin bat izanez.

- Igorleak 3 eta 12 V-eko tentsioan lan egiten du, SAW osziladore batekin, ASK/OOK modulazioarekin, 433.92 MHz maiztasuneko operaketarekin, 25mW potentziarekin 12 V-etan, errore maiztasuna +/- 150 Khz maximoetan eta 10Kbps baino gutxiagoko transmizio abiadura.



7.irudia: RF433 igorlea.

http://mla-s1-p.mlstatic.com/modulo-rf-transmisor-y-receptor-433-mhz-arduino-robotica-pic-20271-MLA20186866356_102014-F.jpg -tik aterata

- Hartzailea 4,5 V eta 5,5V-eko tentsioan lan egiten du, 5,5mA-ko intentsitatearekin, Receptor Superregenerativo hasierako funtzionamenduan, ASK/OOK modulazioan, 433.92 MHz-ko maiztasun operaketarekin, -100dBm-eko sentsibilitatearekin eta 9,6 Kbps baino gutxiagoko transmizio abiaduran.



8.irudia: RF433 hartzailea.

http://mla-s1-p.mlstatic.com/modulo-rf-transmisor-y-receptor-433-mhz-arduino-robotica-pic-20271-MLA20186866356_102014-F.jpg -tik aterata

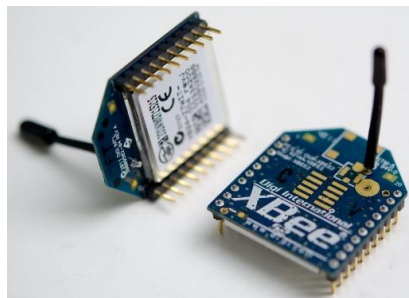
4.2.2. XBEE

XBee-ak errendimendu altuko irrati frekuentziazko moduluak dira. Digi International fabrikatzaileak eginiko modulu hauek kontsumo txikia dute eta prezioz merkeak dira. XBee-ak haririk gabeko konexioa eskeintzen dio proiektuari. Nahiz eta, beste fabrikante batzuen moduluekin bategarria izan, erabili diren moduluak berdinak izan dira.



Modulu hauek ZigBee protokoloaren operatetak egiteko diseinatuta daude, kostu gutxikoak eta energia kontsumo txikikoa izateko. Hori bai, XBee-ak energia kontsumo minimo bat behar dute moduluak datuak elkar trukatzeko.

ZigBee haririk gabeko komunikaziozko protokoloa da, IEEE_802.15.4 haririk gabeko sare komunikaziozko protokoloan oinarritua. Modulu hauek mikrokontroladoreentzat edo portu seria duten beste edozein dispositiboarentzat, komunikazio sinplea eta segurua eskeintzen du.



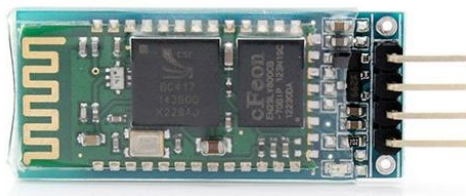
9.irudia: XBee.

https://upload.wikimedia.org/wikipedia/commons/6/63/Pair_of_XBee_Series_2s_with_Whip_Antennas.jpg-tik

aterata

4.2.3. BLUETOOTH

Bluetooth HC-06 modula UART RS 232 serial protokoloa erabiltzen du. Txartelak 4 pin osatutako adaptadore bat dakar, protoboard plakan erraz jartzeko. Board-eko pinak hauek dira: VCC, GND, RX eta TX. Gainera erregulatzailer integratu bat dauka bere elikadura 3,6 V-etik 6 V-era aldatzen duena. Bluetooth V2.0 protokoloarekin bateragarria da. 10 metroko ahalmena dauka.



10.irudia: Bluetooth HC-06.

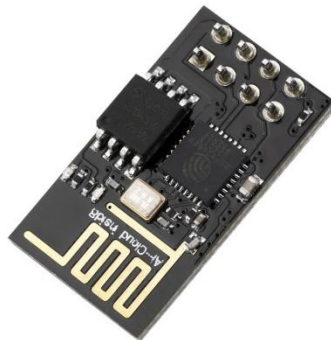
<http://electronilab.co/wp-content/uploads/2013/03/1129201-1.jpg> -tik aterata



4.2.4. WIFI

Wifi Seriak ESP8266 modulua soluzio osoa eskeintzen du, WIFI sare sistementzako konexio ekonomikoa, diseinatzaileari WIFI-z erlazionatutako funtzioak eskuordetuz eta TCP/IP prozesatzaileari aplikazio egiten utziz. Wifi serial ESP8266 moduloa, sare adaptadore bezala lan egin dezake, UART komunikazioa erabiliz.

Modulu berri hau, lehen ez zegoen programatzeko modurik, Arduino IDE programa ez zegoen prestatu eta gainera WIFI komunikazioa kudeatzeko AT komandoen bitartez egin behar da. Domotika aplikazioetan oso erabilia da baina komunikazioa egoteko WIFI sare bat behar da.



11.irudia: WiFi Serial ESP8266.

<http://g03.a.alicdn.com/kf/HTB1L.jHKFXXXXcmXFXXq6xXFXXI/New-Remote-font-b-ESP8266-b-font-Wireless-Module-Stable-font-b-WIFI-b-font-to.jpg> -tik aterata

4.3. KANPO BEHATOKIARENTZAKO ELIKADURA

Kanpo behatokiko plaka behar bezala funtzionatzeko eta bere konponente elektroniko guztiak martxan jartzeko, plaka elikatu behar dira. Behatoki honetako konponenteak, hau dira; sentsorea, XBee-a eta arduinoa. Konponente hauek funtzionamenduan jartzeko, gutxienez 3,3 V-z elikatu behar dira.



4.3.1. NIKEL-METAL BATERIA

Bateria kargagarri hauek koste gutxikoak dira, eta euren dentsitatea LiPo bateriena baino txikiagoa da. Normalki, energia kapazitate gutxi behar duten konponente elektronikoak funtzionamendua jartzeko erabiltzen dira. Bateria hauek 1,2V edo 1,5V-eko tentsio nominala ematen dute.



16.irudia: Nikel-Metal bateriak.

https://upload.wikimedia.org/wikipedia/commons/e/e7/NiMH_2500mAh.jpg -tik aterata

4.3.2. ALKALINA BATERIAK

Bateria hau, ohikoena da eta mota ezberdinekoak daude. AAA motatakoak 1,2 V-eko tentsioa ematen dute. Beste bateria batzuen artean, 9 V-eko tentsioa ematen dutenak aurkitzen dira, hauei, "petaca" izenarekin ezagutzen dira.



17.irudia: Alkalina bateriak.

https://upload.wikimedia.org/wikipedia/commons/2/22/Pila_Duracell_OEA.jpg -tik aterata



4.3.3. SARETIK

Kasu honetan, tentsioa sare elektrikoetik hartzen da, orduan transformadore bat edo eman beharreko tentsioa ematen duen elikadura iturri bat behar da. Honek esan nahiko du, kableen beharra izango dela.



18.irudia: Sare kablea.

https://pixabay.com/p-159709/?no_redirect –tik aterata

4.3.4. LIPO BATERIAK

Merkatuan dauden batería guztietatik, energia dentsitate gehien eskeintzen duena da, deskarga tasa handia eskeiniz ere eta beste motako bateriekin konparatuz tamaina txikiena duena. Bateria bakoitzak, 3,7 V-eko tentsioa ematen du, 4,2 V-eko tentsio maximoa eta 3,0 V-eko tentsio minimoa. Bateria 3 V-eko tentsio minimoa baino gutxiago izanez gero, bateria suntzitu daiteke.



19.irudia: 3,7 V-eko LiPo bateria.



5. AUKERATZEKO IRIZPIDEAK

Lehen aipatutako aukera guztiaetatik, konponente elektroniko egokia aukeratzeko erabiliko diren irizpideak, hurrengo puntu hauetan aipatzen direnak izango dira.

5.1. PREZIOA

Balioa edo proiektu honen balioa kontuan hartzeko puntua da. Aukerarik merkeena aukeratzeko helburuarekin baina GAL-ren helburuak aldatu barik.

5.2. KONTSUMOA

Aukeratu den bateriak, denbora luze batean deskargatu barik egon daiten, konponente ezberdinen kontsumoa begiratzen da. Kontsumo gutxiko konponenteak aukeeratuz, bateriak bizitza luzeagoa izango du. Gaur egun, kontsumo ideiak, garrantzi handia hartu egin du.

5.3. TENTSIO NOMINALA

Baterien kasurako, konponente elektroniko ezberdinek behar duten tentsio nomila jakitea behar da. Aukeratzen den bateria, konponente guztiak elikatu behar ditu. Gerta daiteke zirkuituak jasan beharreko tentsiora ez heltzea eta hori aurrera ateratzeko, batería bi seriean jartzen dira.

5.4. KARGATZEKO AHALMENA

Behatokian erabilitako bateriak kargatzeko ahalmena baldin badauka, puntu garrantzitsua izango du. Bateria aldatzea ez da behar izango edo eskura izatea karga berri eta erraz bat egiteko.



5.5. KOMUNIKAZIO TARTEA

Behatoki bat, beste batekin komunikazioan jartzean, jakin behar da komunikazio tarte maximoa zenbatekoa den. Gerta daiteke tarte maximo hori kontuan ez izatea eta datuak heldu behar diren lekura heldu ez izatea.

5.6. FUNTZIONAMENDUA

Konponente bakoitzaren gaitasuna neurtzen du, funtzio/ekintza aginduak betetzeko dituen trebezia neurtzen du.

5.7. ERABILGARRITASUNA

Sisteman aldaketak egiteko aukera ematen duena eta programa handitzeko aukera ere eskeintzen duena.

5.8. TAMAINA

Tamainaz berba egitean, kontuan izan behar dena da, behatoki bakoitza egiterakoan konponente handiegiak ez direla jarri behar. Plaka pisu gehiegi ez izateko aholkatzen da, eta mugitzeko erraztasunak izateko tamaina txikikoak izatea ere aholkatzen da.



5.9. HAUTAKETA TAULAK

Taula 1: Hautaketa irizpideak.

IRIZPIDEAK	LABURDURA
Prezioa	PRE
Kontsumoa	KON
Tentsio nominala	TENT
Kargatzeko ahalmena	KAR
Komunikazio tarte	KOMU
Funtzionamendua	FUNT
Erabilgarritasuna	ERAB
Tamaina	TAM

Taula 2: Plaka mikrokontroladore hautaketa

Plaka mikrokontroladorea	TAM	PREC	ERAB	FUNT	TENT	KON	KOMU	TOTAL
Peso (%)	15	25	30	5	10	10	5	100
Arduino ProMini	10	8	9	8	6	6	8	820
Freaduino UNO	8	6	6	6	8	8	8	680
Raspberry Pi A+	7	8	5	7	7	6	3	635
Arduino YunMini	8	8	9	8	7	7	6	845
Raspberry Pi 2B	7	7	4	6	8	8	3	605
Intel Edison	7	5	4	6	9	9	3	575

Taula 3: Haririk gabeko komunikazio hautaketa

Haririk gabeko komu.	TAM	PREZ	TENT	ERAB	KAR	FUNT	KOMU	TOTAL
Peso(%)	10	10	20	15	10	15	20	100
RF433 Mhz	8	9	7	6	7	6	7	750
XBee	6	5	9	9	9	7	9	805
Bluetooth HC-06	8	7	5	4	4	4	6	530
Wifi ESP8266	8	6	6	5	5	5	6	645



Taula 4: behatokiko elikadura hautaketa

Behatokiko elikadura	TAM	PRE	KAR	KON	TENT	TOTAL
Pisua (%)	15	15	20	25	25	100
Nikel-metal	4	5	4	6	9	640
Alkalina	4	8	3	6	9	570
LiPo 3.7V	10	5	9	8	9	835
Sarea	3	9	2	10	9	705



Taula 5: Arduino eta sentsore konexio deskribapena.

Arduino YunMini	Kable kolorea	HTU21D sentsorea
3,3 V	Urdina	3,3 V
GND	Beltza	GND
2	Berdea	SDA
3	Gorria	SDL

Sentsorea, arduino programan programatzeko, bere liburutegia sartu behar zaio. HTU21D sentsorearen liburutegiaren izena <SparkFunHTU21D.h> da. Liburutegi hau, bere barneko karpetan adibide bat du. Bere adibidea erreferentzia bezala harturik programatzen hasteko aukera dago arduino YunMini-a eta HTU21D sentsorearen artean. Sentsorea programatzeko orduan, agindu batzuk izan behar dira kontuan:

- Hasieran sentsorea definitu: *HTU21D sentsorea*;
- Sentsorea abiarazi egin behar da: *sentsorea.begin()*;
- Ondoren, sentsoreari hezetasuna eta tenperatura neurtzeko agindua idazten da eta balioa aldagai batean gordetzen da. Balioa hamartarra izan daitekenez float aldagai batean gordeko da arduinoaren memorian:

```
float hezeB = sentsorea.readHumidity();
```

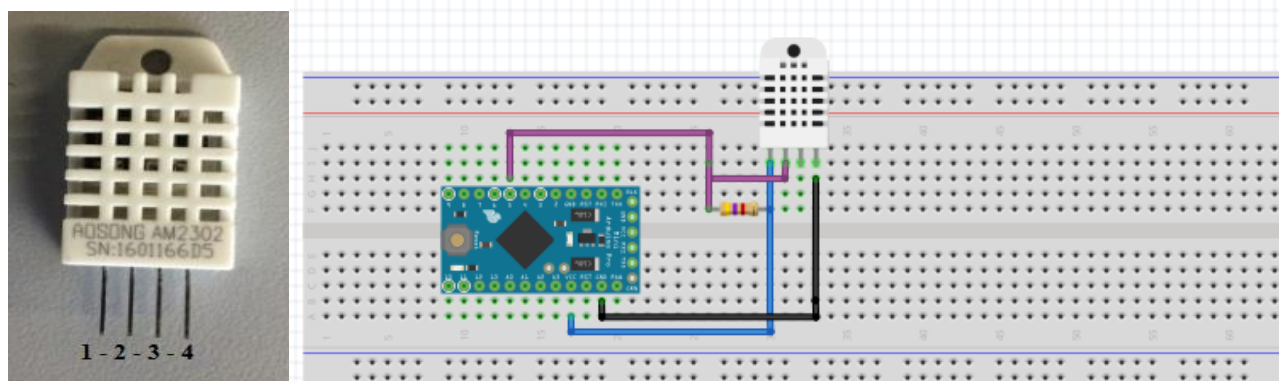
```
float tenpB = sentsorea.readTemperature();
```

HTU21D sentsoreak neurtutako balioak bi aldagaietan gordeta geratu dira: *hezeB* eta *tenpB* aldagaietan. Aldagai horietako datuak ikusi nahi direnean egin beharreko gauza bakarra da, aldagai horiei deitzea izango litzake.



6.1.2. KANPOKO SENTSOREA

Kanpoko behatokian aukeratu den sentsorea aldiz, AM2302 sentsorea da edo DHT22 izenarekin ezagunagoa dena. Sentsore honek ere, ez ditu programatzeko arazo handirik sortzen eta gainera, merkea da ere. Sentsore honek sortu duen arazorik nabarmenena izan da, protoboard plakan DHT22 sentsorea jartzerakoan, hanken lodiera oso eztua zela protoboard plakako zuloetan sartzeko eta horren erruz, DHT22-ak ez zuen konexio ona egiten. Arazo hori aurrera ateratzeko egin dena izan da, sentsoreari hanka lodiago batzuk soldatzea protoboard plakan hobeto konektatu daiten. DHT22 sentsore honek, konexioak egiteko lau hanka ditu. Lehenengo hankan 3,3 V eta 6 V-en arteko tentsioz elikatu behar dela dio bere datasheet-a. Laugarren hanka lurrera konektatu behar zaio eta bigarren hanka arduinoko bostgarren pin batera. Baita, erresistentzia bat jarri beharko da tentsio hankaren eta OUTPUT hankaren artean. Erabiliko den erresistentzia 4,7k Ω -eko izango da.



21.irudia:DHT22 pin-etako argazkia eta arduinoarekin duen konexioa irudia.

DHT22 sentsoreak arduino programan programatzeko bere liburutegia behar du. Sentsore honen liburutegia <DHT.h> da. Barneko behatokian erabili den sentsorearen antzera programatzen da.

- Lehenik sentsorea definitu: zein pin-etan konektatu den eta ze motakoa sentsorea den.

```
#define DHTPIN 5
#define DHTTYPE 22
DHT dht(DHTPIN, DHTTYPE);
```




- Bestean egin den bezala sentsorea abiarazi egingo da:

```
dht.begin();
```

- Balioak aldagaietan gordeko dira arduinoaren memorian gordeta izateko, jakinik zenbakiak hamartarrak izan daitezke:

```
float hezeK = read.Humidity();
```

```
float tenpK = read.Temperature();
```

Kanpoko behatokiko datuak arduino memorian gordeta geratuko dira: hezeteasunaren datua hezeK algaian eta tenperaturaren datua tenpK algaian. Datuak ikusi nahi direnean egin beharreko gauza bakarra, aldagai horiei deitzea izango da.

Taula 6: Arduino eta sentsore konexio deskribapena.

Arduino ProMini	Kolorea	DHT22 Sentsorea
Vcc	Urdina	1
5	Morea	2
-	-	3
GND	Beltza	4

6.2. XBee-ak

XBee-n helburu nagusia, elkar komunikatzea da. Kanpo behatokiko datuak, barne behatokian irakur daitezela. XBee-en arteko komunikazioa konfigutatu aurretik, XBee bakoitzari XBee Adapter Board bat jarriko zaio. XBee adaptadore honeri esker, USB portua konektatuz, XBee-aren konfigurazioa egiten hasi daiteke.



6.2.1. XBee Adapter Board eta USB portuaren arteko konexioa

XBee txip-ak plakan jarri aurretik, XBee Adapter Board #32403 bat jarri behar zaio. Adaptadore honen betebeharretako bat da, 5 V-eko tentsioa 3,3 V-era pasaraztea da. Beste bat adibidez, XBee-aren pin-ak protoboard plakan kokatu al izango direla, zeren eta XBee-aren pin-ak bata bestetik oso gertu daude eta protoboard-eko zuloekin ez dira egokitzen. XBee adaptadore honen helburu nagusia da, XBee bakoitzari bere konfigurazioa egitea uztea. Honeri esker, USB mini-aren eta XBee adaptadorearen arteko konexioa egin daiteke eta orduan, XBee-ak konfiguratzeko prest egongo liratezke. XBee bakoitzari USB mini ezberdin bat konektatu zaio.

Figure 1-03. XBee®/XBee-PRO® RF Module Pin Numbers
(top sides shown - shields on bottom)

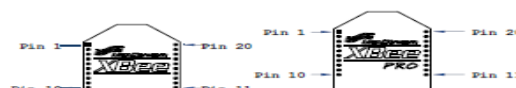


Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / <u>CONFIG</u>	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	<u>RESET</u>	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / <u>SLEEP</u>	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	<u>RTS</u> / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

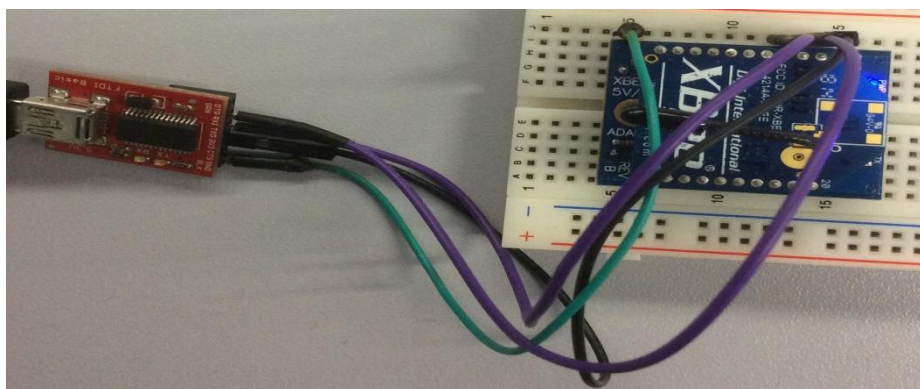
22.irudia: XBee adapter pin-ak.



6.2.2. Kanpo behatokiko XBee Adapter Board eta USB portuaren arteko konexioa

Kanpo behatokiko XBee-a ordenagailuan konfiguratzeko, USB mini bat erabilia egingo da. Bere komunikazioa, portu serie bitartez egingo da.

- Kanpo behatokian, USB mini FTDI basic (Sparkfun) eta Xbee Adapter Board #32403 arteko konexioa egin da.



23.irudia: FTDI eta XBee konexioa.

Taula 7: USB eta XBee adapter konexioa.

USB mini FTDI basic	Kolorea	XBee Adapter Board
3,3 V	Beltza	2 (Vcc)
GND	Berdea	11 (Vss)
RX	Morea	3 (Dout)
TX	Morea	4 (Din)



6.2.3. Barne behatokiko XBee Adapter Board eta USB portuaren arteko konexioa

- Barne behatokian aldiz, USB mini-B FT232R eta Xbee Adapter Board #32403 arteko konexioa egin da.

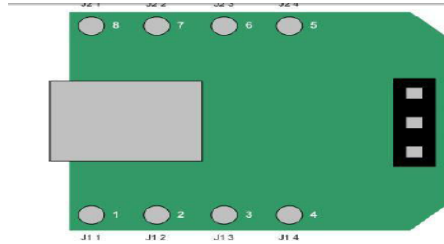


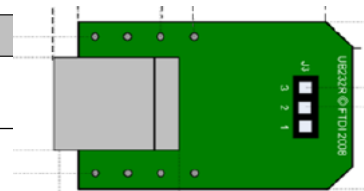
Figure 3.1 UB232R Electrical Connections (Top View)

Silkscreen Pin	Connector Pin	Name	Description
1	J1-1	GND	0V Power pin
2	J1-2	VCC	+5V Power from USB Interface
3	J1-3	CTS#	FT232R CTS pin
4	J1-4	RTS#	FT232R RTS pin
5	J2-4	CBUS1	FT232R CBUS1 pin
6	J2-3	CBUS0	FT232R CBUS0 pin
7	J2-2	RXD	FT232R RXD pin
8	J2-1	TXD	FT232R TXD pin

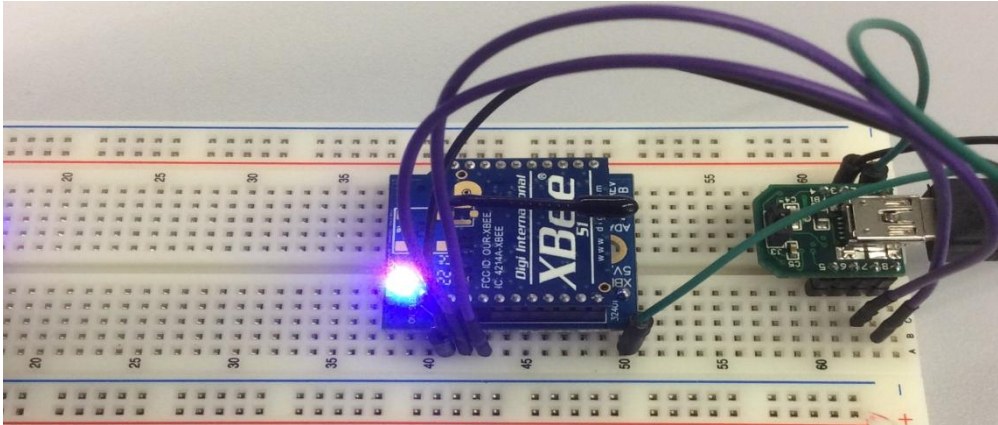
24.irudia: FT232R-aren pin-ak eta pin-en deskribapena.

- Kontuan izan behar da USB mini-B FT232R honetan, J3 konektorea nola dagoen jarrita (J3 izena goian dagoela ikusiz, J3 konektoreko pinak honako segida dute: 3, 2, 1).
 - 1 eta 2 pin-ak konektaturik badaude 5 V-ekin egingo du lan.
 - 2 eta 3 pin-ak konektaturik badaude 3,3 V-ekin egingo du lan.

Connector J3	Function
Link 1-2 connected	VCCIO and RESET# driven connected to VCC (+5V)
Link 2-3 connected	VCCIO and RESET# driven connected to +3.3V output of FT232R LDO regulator



25.irudia: FT232R, J3 konexio pin-ak.



26.irduia: FT232R eta XBee konexioa.

Taula 8: USB eta XBee adaptadore konexio deskribapena.

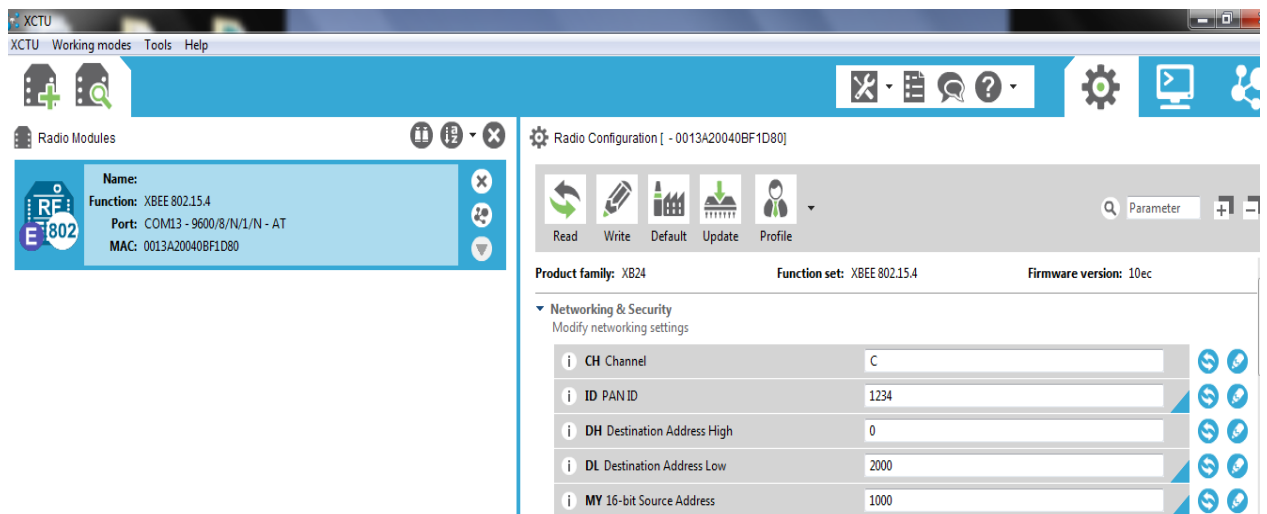
USB mini-B FT232R	Kolorea	XBee Adapter Board
1	Berdea	11.pin-a (Vss)
2	Beltza	2.pin-a (Vcc)
7	Morea	3.pin-a (Dout)
8	Morea	4.pin-a (Din)



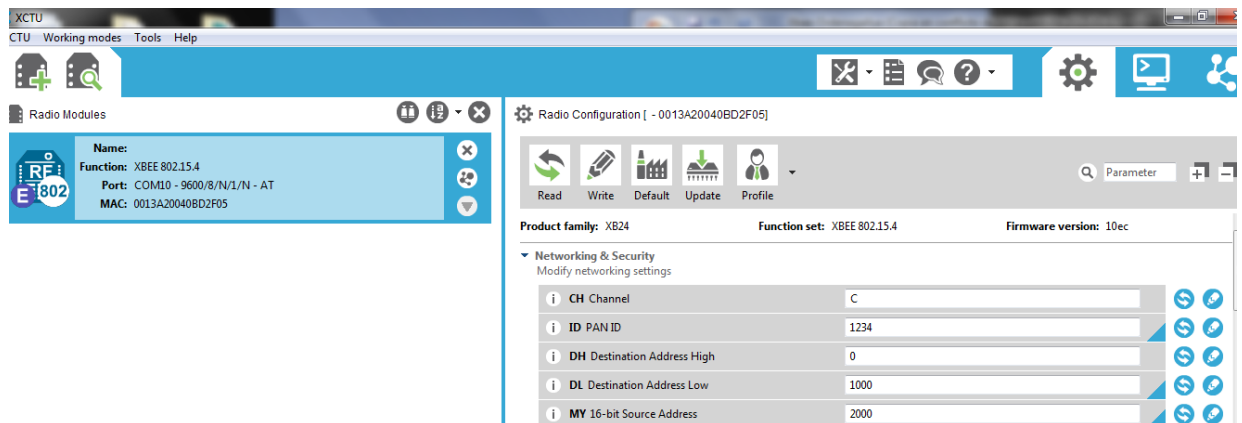
6.2.4. XBee-ak konfiguratzeko eta egiaztatzeko programa

XBee-ak ordenagailuarekin konektatuta daudenean, XCTU programa zalbaldu egingo da. Programa hau, kasu honetan, XBee-ak konfiguratzeko erabiltzen da. XBee-ak euren arteko elkar komunikazioa betetzeko, datu zehatz batzuk sartu behar zaizkio XBee bakoitzari. XBee bakoitzean idatzi beharreko datuak ondorengoak dira:

- **ID PAN ID** (Zenbaki bera bi XBee-etan)
- **MY 16-bit Source Address** (XBee bakoitzari zenbaki ezberdin bat)
- **DL Destination Address Low** (elkar trukatu nahi dugun XBee-aren **MY** zenbakia)



27.irudia: Barne behatokiko XBee-aren datuak.



28.irudia: Kanpo behatokiko XBee-aren datuak.

USB mini FTDI basic konektatzean kanpo behatokiko XBee-arekin, arazoak eman ditu. Orduan, XBee-a konektatzeko erabili egin den USB-a, beste USB mini-B FT232R bat izan da. Azkenean bi XBee-ak X-CTU programan konfiguratzeko, bi USB mini-B FT232R erabili dira behatoki bakitzekeo XBee-entzat.

X-CTU programan, XBee bakoitzari jarri beharreko datuak jarri ondoren, X-CTU programatik irten barik, konsola aukera erabilia, elkar komunikatzen direla konprobatu daiteke. XBee baten konsola pantailan mezu bat idatziz, beste XBee-aren konsola pantailan mezua agertu beharko da. Berdina erantzun egin nahi denean, mezua bi XBee-etan agertuko beharko da konfigurazioa ondo egin dela frogatzeko.

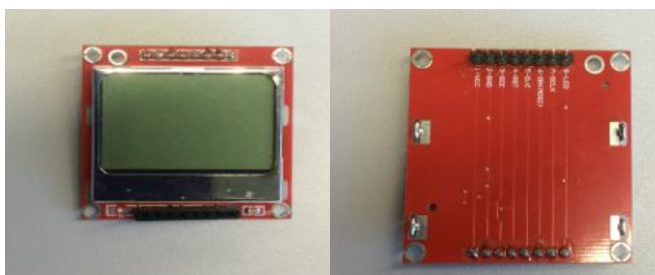
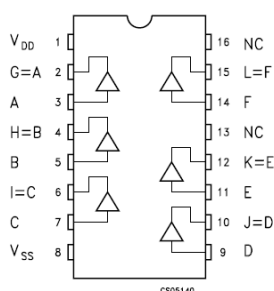
Bi XBee-etan bata bestearen mezua agertzen direnean, elkar komunikatzen ari direla esan nahi du, eta prest daudela zirkuituan konektatzeko datuak elkar trukatzeko.



6.3. PANTAILA

Barneko behatokian datuak ikusteko pantaila bat jarriko da, eta aukeratu egin den pantaila, PCD8544 nokia LCD display-a izan da. Pantaila honetan erakutsi egingo diren datuak izango dira; alde batetik, kanpo behatokiko hezetasuna eta tenperatura eta bestetik, barne behatokiko hezetasuna eta tenperatura ere. PCD8544 nokia, pantaila honek energia kontsumo txikia dauka; backlight pina erabili gabe izten baldin bada bere kontsumoa 5mA baino txikiagokoa izango da eta backlight pina erabiliz gero 20mA-ko kontsumoa izango da gutxigorabehera. LCD pantailaren bereizmena 48x84 pixelekoa da. Pantailak lan egiteko 3,3 V-eko tentsioa behar du, baina arduino YunMini-ak 5 V-era lan egiten du. Arduino YunMini-a 3,3 V-eko tentsioa emateko pin bat dauka baina beste pin guztiak 5 V-etan egiten dute lan, orduan arazo horrerri aurre egiteko maila bihurgailu bat jarri egin da.

Aukeratu egin den maila bihurgailua 5040-a izan da. Maila bihurgailu honek 3,3 V-eko tentsioa 5 V-etan bihurtzen du eta berdina kontrako bidea artzen badu. Maila bihurgailuari Vdd tentsioa sartu behar zaio, hau da 3,3 V-eko tentsioa eta Vss pin-ean GND konektatu behar zaio, hurrengo pinetan aldaketa tentsioa prest izateko eta erabiltzeko arazorik ez izateko. Orduan, 2, 4, 6, 10, 12 eta 15 pin-ek 3,3 V-eko tentsioa izango dute eta 3, 5, 7, 9, 11 eta 14 pin-ek, aldiz, 5 V-eko tentsioa izango dute.



29.irudia: Maila bihurgailu eta pantaila pin-ak.

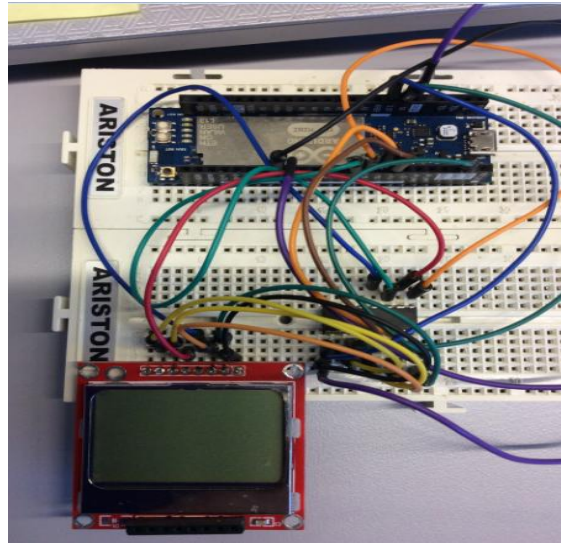


Arduinoa, maila bihurgailua eta LCD pantailaren arteko konexioak hurrengokoa taulan deskribatzen dira:

Taula 9: Arduino, maila bihurgailu eta pantaila konexio deskribapena.

Arduino YunMini	Maila bihurgailua		Pantaila
	5 V	3,3 V	
3,3 V	1	1	Vcc (1)
GND	8	8	GND (2)
CS (6)	7	6	SCE (3)
RST (4)	11	12	RST (4)
D/C (5)	9	10	D/C (5)
MOSI	5	4	MOSI (6)
SCK	3	2	SCLK (7)

Pantailaren LED pin-a konektatu nahiko bazen, arduinoko tentsio igorletik kable bat konektatu beharko zen. Konexioa egitean, pantaila led batzuen bidez argiago ikusten da pantaila. Led horiek kontsumo handirik ez emateko eta argi maila normal bat emateko pantailari erresistentzia bat jartzen zaio.



30.irudia: Barne behatokiko arduino, pantaila eta maila bihurgailu konexioak.

Arduino programan LCD pantaila programatzeko eta nahi diren datuak pantailan inprimatzeko, lehenik pantaila prestatu egin behar da:

- Lehenengo pausoa izan da, pantailaren liburutegia programan sartzea, <"U8glib.h"> liburutegia hain zuzen ere.
- Pantailaren hardware portuak definitu egin behar dira eta aukeratu beharreko pantaila definitzen da, kasu honetan PCD8544 pantaila da. Pantailari *u8g* izena jarri zaio eta definitu beharreko hardware portuak hurrengoak dira: cs = 6, D/C = 5 eta RST = 4.

PCD8544, 84x48

- Tested displays: Nokia display
- Arduino C++ Interface

Description	U8glib Constructor
SW SPI	U8GLIB_PCD8544(sck, mosi, cs, a0 [, reset])
HW SPI	U8GLIB_PCD8544(cs, a0 [, reset])

31.irudia: Pantailaren datasheet datuak.



- *setup* funtzioan pantailari agintzen zaio *setRot180* agindua. Agindu honek dio pantaila 180 graduko posizio aldaketa jasoko duela, zeren eta pantailan ikusi beharreko datuak buruz behera geratzen direlako eta.

```
xbee_pantaila_barruko Arduino 1.6.12
Archivo Editar Programa Herramientas Ayuda
xbee_pantaila_barruko
#include <SoftwareSerial.h>
#include "U8glib.h"
#include "SparkFunHTU21D.h"
#include <Process.h>

SoftwareSerial xbee(8, 9); //Rx, Tx
U8GLIB_PCD8544 u8g(6, 5, 4); // SPI HW com: CS = 6, A0 D/C = 5, Reset = 4
HTU21D sentsorea;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Arduino Yun abiarazi
  //while(!Serial) {}

  Bridge.begin();

  xbee.begin(9600); // Xbee abiarazi
  sentsorea.begin(); // HTU21D abiarazi
  u8g.setRot180(); // Pantailari buelta emateko
}
}
```

32.irudia: Arduino programa.

- *loop* programa nagusian pantailari deituko zaio esanez, lehenengo orrian idazteko prest egoteko. *u8g.firstpage()*.
- Ondoren *do* aginduaren bitartez, *draw* funtziora salto egingo du, izendatutako aldagaiak inprimatzeko. *draw* funtzioko aginduak betetzean, berriz programaren puntu berdinerara itzuliko da eta *while* baten barruan sartuko da, hurrengo orri salto agindua betetzeko. (*u8g.nextpage()*)
- *draw()* funtzioan, lehenik aldagaiak definitzen dira zenbaki osoak ez diren bezala.
- Pantailan inprimatu beharreko karakterezko datuak idazteko agindua hau bidaltzen da *setFont()*, eta aukeratuko den letra mota ondoren, *u8g_font_coursB08* .
- Karakterezko datuak idazteko beharrezkoa den agindua *drawStr()* da eta bere barnean X eta Y posizioak definituko dira eta ondoren, idatzi nahi dena idatziko da.



- Lerro bat marraztu nahi denean idatzi beharreko agindua *drawLine()* da. Bere barnean definitu beharreko datuak dira X eta Y zein posizioan hasten diren eta zeinetan bukatzen duten.
- Nahi den datua idazteko, lehenik posizioa definitzen da *setPrintPos()* aginduarekin eta bere barnean posizioa zehaztuz.
- Posizioa zehaztuta dagoenean *print()* agindua agintzen zaio, inprimatu nahi den datuaren aldagaiari deituz eta nahi diren dezimal kopuruak eskatuz.

```
xbee_pantaila_barruko Arduino 1.6.12
Archivo  Editor  Programa  Herramientas  Ayuda

xbee_pantaila_barruko

// picture loop
u8g.firstPage(); // Pantailako lehengo orrian idazteko
do {
  draw(hezeK, tenpK, hezeB, tenpB); // Pantailan idatzi beharreko aldagaiak
} while( u8g.nextPage() ); // Hurrengo orrira salto egiteko

// datuen bidalketa hodeira
uploadData2Cloud(hezeK, tenpK, hezeB, tenpB);
}

void draw(float hezeK, float tenpK, float hezeB, float tenpB) {
  u8g.setFont(u8g_font_courB08); // Pantailako letra mota
  u8g.drawStr(11, 13, "INT."); // Pantailan idatzi beharrekoa eta bere posizioa
  u8g.drawStr(55, 13, "EXT."); // Pantailan idatzi beharrekoa eta bere posizioa
  u8g.drawLine(44, 0, 44, 48); // Pantailan marrazteko
  u8g.drawLine(0, 14, 84, 14); // Pantailan marrazteko
  u8g.setPrintPos(4, 30); // Datorren datuaren posizioa
  u8g.print(tenpB, 1); // Ze barruko datu idatzi eta bere dezimalak
  u8g.setPrintPos(4, 45); // Datorren datuaren posizioa
  u8g.print(hezeB, 1); // Ze barruko datu idatzi eta bere dezimalak
  u8g.setPrintPos(48, 30); // Datorren datuaren posizioa
  u8g.print(tenpK, 1); // Ze kanpoko datu idatzi eta bere dezimalak
  u8g.setPrintPos(48, 45); // Datorren datuaren posizioa
  u8g.print(hezeK, 1); // Ze kanpoko datu idatzi eta bere dezimalak
}
```

33.irudia: Arduino programa.



7. BEHATOKIAK

Bi behatokiak, XBee-i esker, elkar komunikatzen direla jakin da, arduinoa konektatzeko momentua heldu da. XBee bakoitzari arduino ezberdin bat konektatuko zaio. Kanpo behatokian, XBee-ak arduino ProMini bat izango du konektatua eta barne behatokian konektatuko zaion arduinoa, YunMini-a izango da.

PROIEKTU HONEN INGURUAN EGINIKO KODE GUZTIA GITHUB-EKO BILTEGI BATEAN GORDETA DAGO, BERE ESTEKA ONDORENGOA IZANIK:
https://github.com/bcastrillo/eguraldi_behatokia

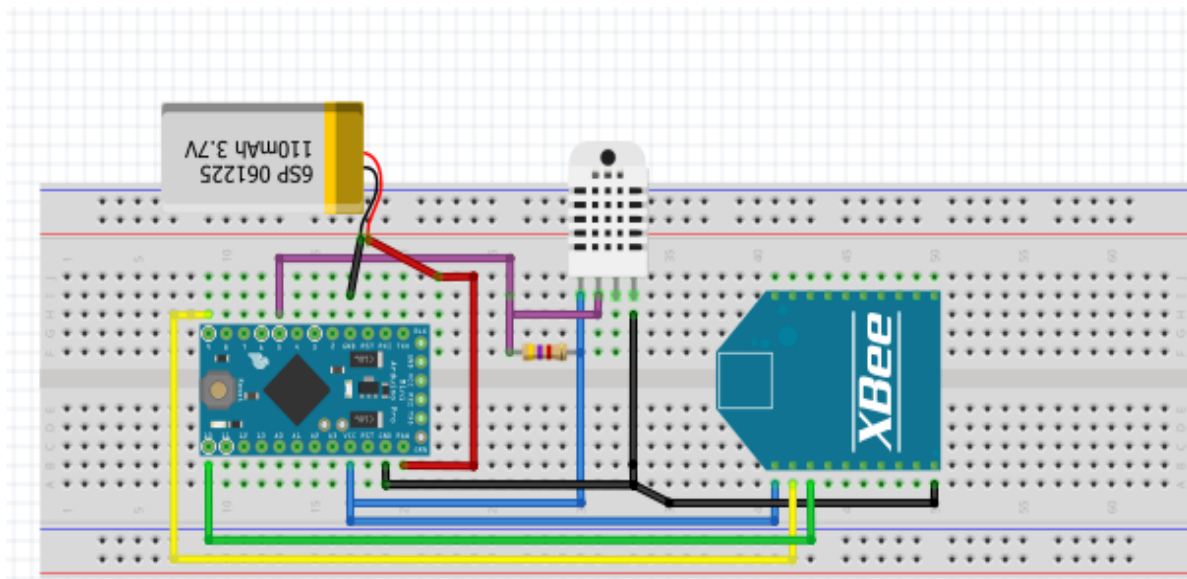
7.1. KANPOKO BEHATOKIA

Lehenik, kanpo behatokiko muntaiaren konexioak eta programazioa deskribatu egingo da. Behatoki honetan egon diren konponenteak elektronikoak hurrengokoak izango dira: arduino ProMini-a, XBee-a, DHT22 sentsoarea eta 3,7 V-eko LiPo batería bat.

Arduino ProMini honek, DHT22 sentsoareko datuak memorian gordeko ditu eta XBee-ari pasatuko dizkio, bidaltzeko prest izateko. Behatokia hau elikatzeke, batería batez elikatu egingo da.

7.1.1. Kanpo behatokiko konexioak

Kanpo behatokian izango diren konponente elektronikoen arteko konexioak hurrengoko irudian erakutsi egingo dira. Konponente guztiak, arduino ProMini-ari konektaturik egongo dira.



34.irudia: Kanpo behatokiko konexioak.

Taula 10: Arduino ProMini eta XBee adaptadore konexio deskribapena.

Arduino ProMini	Kolorea	XBee Adapter Board
Vcc	Urdina	2.pin-a (Vcc)
GND	Beltza	11.pin-a (Vss)
9.pin-a (Rx izango da)	Horia	3.pin-a (Dout)
10.pin-a (Tx izango da)	Berdea	4.pin-a (Din)

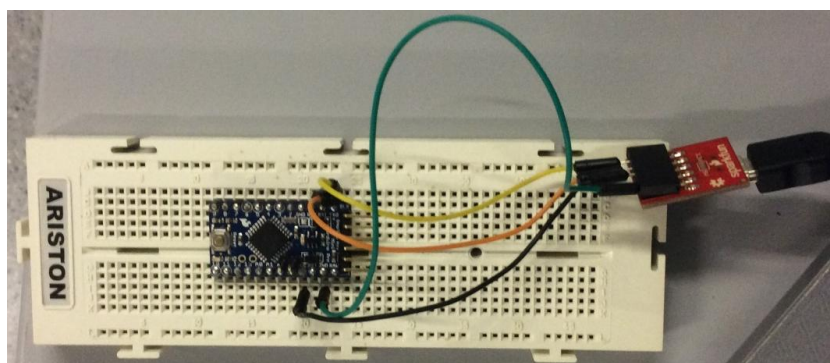
Taula 11: Arduino ProMini eta bateria konexio deskribapena.

Arduino ProMini	Kolorea	Bateria
RAW	Gorria	(+) Gorria
GND	Beltza	(-) Beltza



7.1.2. Kanpo behatokiko programazioa

Kanpoko behatokiko konexio guztiak eginda daudenean, arduinoaren programa burutzea falta izango da. Arduino ProMini-ari programa igotzeko, USB bat konektatu behar zaio. Arduino ProMini-a, arduino programara konektatzeko USB mini FTDI basic Sparkfun-ekoa erabiliko da. Kasu honetan, ez da egon arazorik arduinoa eta USB-a konektatzean arduino programarekin nahiz eta, lehen arazoak izan diren XBee-a eta USB-aren artean konexioan X-CTU programarekin. Arduino ProMini-a eta USB mini FTDI basic Sparkfun-eko konektorea elkar konektatzeko kableen bidez egingo da. USB-a arduinoaren hankei jarraian konektatzean, ez du ondo antzematen eta arazoak ematen ditu. Horregaitik kableen bidez konektatzea erabaki da, hurrengo konexioa eginez:



35.irudia: Programa arduinora igotzeko konexioa.

Taula 12: Arduino ProMini eta USB konexio deskribapena.

Arduino ProMini	Kolorea	USB mini FTDI basic
Vcc	Beltza	3,3 V
GND	Berdea	GND
RX	Laranja	TX
TX	Horia	RX



Kanpoko behatokiko programaren helburua da, sentsoreak neurtzen dituen datuak arduinoko memorian gordetzea eta honek, datu horiek XBee-ari pasaraztea, bidaltzeko prest izateko. Helburua argi izan da, falta dena programa burutzea da.

- Hasieran azaldu den bezala, lehenik hezetasuna eta tenperatura neurketak algaietan gorde egingo dira.
- *if* funtzio bat jartzen da konprobatzeko ea heldu diren balioak, zenbakiak ez direla ikusteko. Zenbakiak ez badira "Error" erantzungo du, zenbakiak baldin badira aurrera jarraituko du.
- Arduino eta XBee-aren arteko konexioan, arduinoko 6.pin-a eta XBee-aren DTR(9.pina) lokartzeko erabiliko dira. Arduinoko 6.pina desaktibatu egingo da, lo modutik irtetzeko. Agindua horrela adieratzen da:

```
digitalWrite(6, LOW)
```

- Kanpoko behatokiko neurketak, barneko behatokiko pantailan agertu egingo direnez, kanpoko XBee-ari agindu egingo zaio *hezeK* eta *tenpK* aldagaiak, XBee bitartez bidaltzeko. XBee-ak datuak bidaltzeko erabiltzen dituen aginduak, honako hauek dira:

```
xbec.print("hezeK");
```

```
xbec.print(" ");
```

```
xbec.print("tenpK");
```

- Datuak bidali direnean eta XBee-aren memoria garbitzeko agindua erabiltzen da.

```
xbec.flush();
```

- Denbora pixka bat ematen zaio, eta arduinoaren 6. pin-a aktibatzen da, lo moduan sartzeko. Horrela deskribatzen dira agindu hauek:

```
delay (1000);
```

```
digitalWrite(6, HIGH);
```




```
xbee_proba_kanpoko1 Arduino 1.6.12
Archivo Editar Programa Herramientas Ayuda
xbee_proba_kanpoko1

void loop() {
  // put your main code here, to run repeatedly:

  float hezeK = dht.readHumidity(); //HTU21D-tik hezetasuna neurtu eta humd aldagaian gorde
  float tenpK = dht.readTemperature(); //HTU21D-tik temperatura neurtu eta temp aldagaian gorde

  if (isnan(hezeK) || isnan(tenpK)) {
    Serial.print("Error");
    return;
  }

  //float hezeK = 2.0; //Sentsore barik aproba egiteko
  //float tenpK = 1.0; //Sentsore barik aproba egiteko
  digitalWrite(6, LOW);
  delay(50);
  Serial.print(hezeK); Serial.print(" "); Serial.println(tenpK);
  xbee.print(hezeK); // Xbee-tik bidali beharreko aldagaia
  xbee.print(" "); // Xbee-tik bidali beharreko datua
  xbee.print(tenpK); // Xbee-tik bidali beharreko aldagaia
  xbee.flush();
  delay(1000);
  digitalWrite(6, HIGH);
}
```

36.irudia: Kanpo behatokiko arduino programa.

7.2. BARNEKO BEHATOKIA

Orain, barne behatokian doan muntaia deskribatuko da. Behatoki honetan egongo diren konponenteak hurrengokoak izango dira: arduino YunMini-a, XBee-a, HTU21D sentsorea eta pantaila bat.

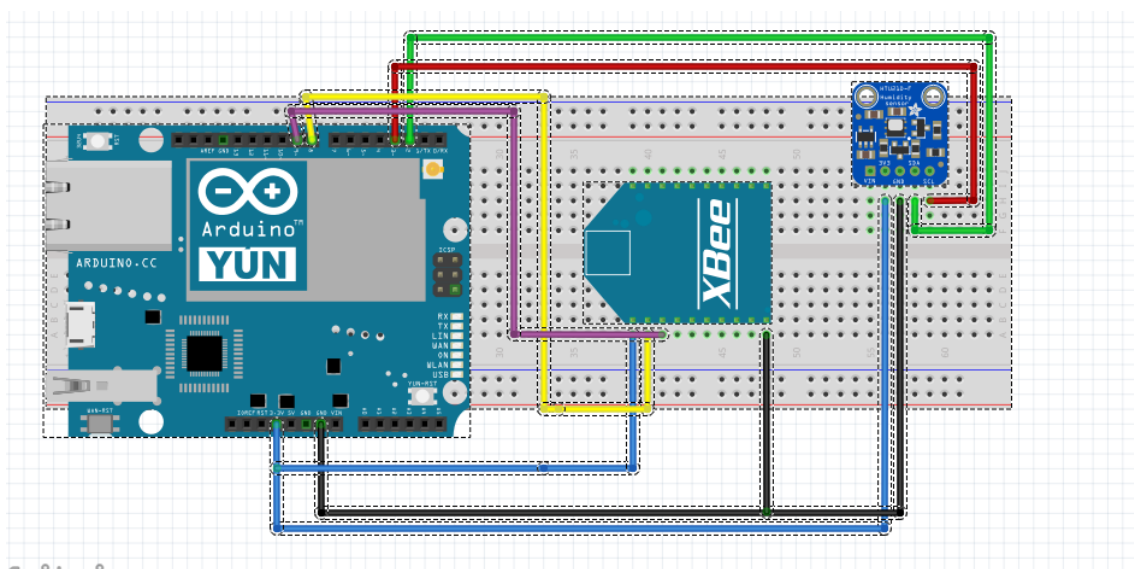
Barne behatokiko arduino YunMini-ari, XBee bat konektatu zaio. XBee honek, kanpo behatokiko XBee-arekin elkar komunikazioa lortu egin da X-CTU programaren bitartez. Barneko behatokiko XBee-aren helburua, kanpoko behatokiko datuak jasotzea da, eta honek, datu horiek arduino YunMini-ko memoriari pasaraztea.

Arduino YunMini honek ere, XBee-tik aparte beste konponente batzuk ditu konektatuta, HTU21D sentsorea adibidez. Sentsore honek, hezetasuna eta temperatura ere neurtzen ditu. Arduinoko memoriako datu horiek pantaila batean inprimatzea faltako zen bakarrik.



7.2.1. Barne behatoliko konexioak

Barneko behatokian erabili diren konponenteen elkarren arteko konexioak, ondorengo irudian erakutsi egingo dira. Barne behatokian erabili den arduinoa, YunMini-a izan da. Arduino honeri konponente elektroniko guztiak konektatu zaizkio. Barne behatoki honetan konponenteen arteko konexioak, oraindik irudikatu ez direnak deskribatuko dira.



37.irudia: Barne behatoliko arduinoa, XBee-a eta sentsorearen arteko konexioak.

Taula 13: Arduino YunMini eta XBee adaptadore konexio deskribapena.

Arduino YunMini-a	Kolorea	XBee Adapter Board
3,3 V	Urdina	2.pin-a (Vcc)
GND	Beltza	11.pin-a (Vss)
8.pin-a (Rx izango da)	Horia	3.pin-a (Dout)
9.pin-a (Tx izango da)	Morea	4.pin-a (Din)



7.2.2. Barne behatokiko programazioa

Barne behatokiko programazioa arduino YunMini-ra pasarazteko, USB mini kable bat behar da. Arduino YunMini honeri, ez zaio USB mini sarrera portu bat jarri behar programa bere memoriara igotzeko. Arduino YunMini-k bere USB portu integratua dauka jadanik. Horri esker, kable eta konponente gehiago aurretik dira nahiz eta, batzuetan arazoak eman USB portuak, arduino programarekin konektatzean. Arduino YunMini-ren USB portua ez dela oso ona esan daiteke, eta horregaitik batzuetan konexio arazoak ematen ditu. Barne behatokia programatzeko erabili diren helburuak honako hauek izan dira:

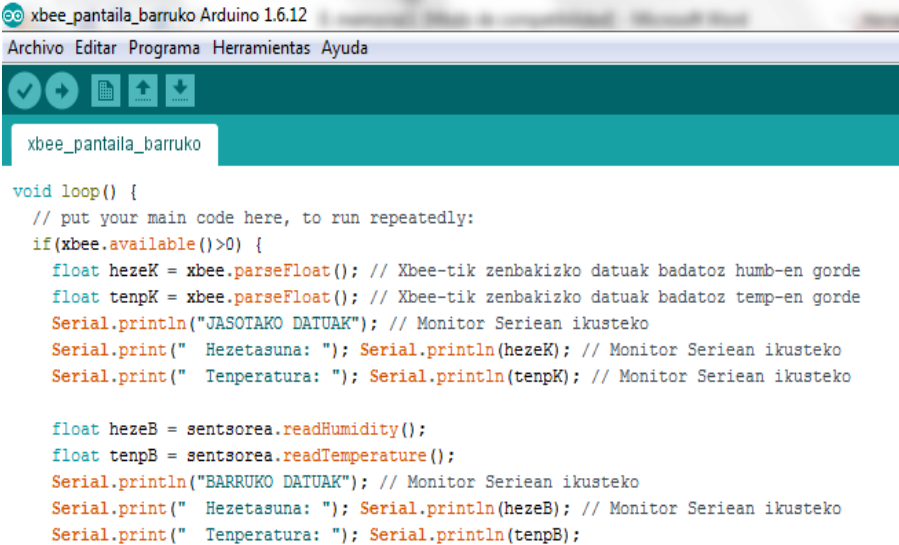
- Kanpo behatokitik datozen zenbakizko datuak, barne behatokiko XBee-aren bitartez datuak irakurtzea eta aldagaietan gordetzea.
- Barne behatokiko hezetasuna eta temperatura, sentsoretik irakurtzea eta aldagaietan gordetzea.

Barne behatokiko programa nagusiaren programazio deskribapena hurrengokoa da:

- Arduino YunMini-a, XBee-a eta HTU21D sentsorea abiarazi egingo dira agindu hauekin:
 - Arduino YunMini → `Serial.begin(9600);`
 - XBee-a → `xbee.begin();`
 - HTU21D sentsorea → `sentsorea.begin();`
- Lehenago azaldu egin da, kanpoko behatokiko datuak zelan lortzen ziren eta zelan XBee-aren bitartez bidaltzen ziren datu horiek. Orain kanpoko datuak barne behatokira heldu direla konprobatzeko, *if* funtzio bat jartzen da esanez ea XBee-an daturik heldu diren edo ez: `if(xbee.available())>0;`
- Datuak barne behatokiko XBee-ari heldu bazaizkio, datuak irakurtzeko agindua emango zaio `xbee.parseFloat()` aginduaren bitartez eta aldagai berri hauetan gordetzeko esango zaio; *hezeK* eta *tenpK*. Datu horiek inprimatu nahi direnean, egin beharreko gauza bakarra aldagai horiei deitzea izango da.
- Konprobatu nahi baldin bada ea zein datu jasotzen diren ikusteko *Monitor Seriean*, egin daiteke arazorik barik `Serial.print("")` agindua erabiliz.



- HTU21D sentsorearen programazioan azaldu den bezala, sentsoreak neurtutako datuak bi aldagaietan gordeko ditu: *hezeB* eta *tenpB*.
- Aldagai horietan zeozer gordetzen ari dela konprobatzeko eta sentsoreak ez duela neurketa kaskarririk egiten ikusteko, *Monitor Seriean* inprimatzeko agindua emago zaio *Serial.print("")* agindua erabiliz.



```
void loop() {
  // put your main code here, to run repeatedly:
  if(xbee.available()>0) {
    float hezeK = xbee.parseFloat(); // Xbee-tik zenbakizko datuak badatoz humb-en gorde
    float tenpK = xbee.parseFloat(); // Xbee-tik zenbakizko datuak badatoz temp-en gorde
    Serial.println("JASOTAKO DATUAK"); // Monitor Seriean ikusteko
    Serial.print("  Hezetasuna: "); Serial.println(hezeK); // Monitor Seriean ikusteko
    Serial.print("  Temperatura: "); Serial.println(tenpK); // Monitor Seriean ikusteko

    float hezeB = sentsorea.readHumidity();
    float tenpB = sentsorea.readTemperature();
    Serial.println("BARRUKO DATUAK"); // Monitor Seriean ikusteko
    Serial.print("  Hezetasuna: "); Serial.println(hezeB); // Monitor Seriean ikusteko
    Serial.print("  Temperatura: "); Serial.println(tenpB);
  }
}
```

38.irudia: Barne behatokiko arduino programa.

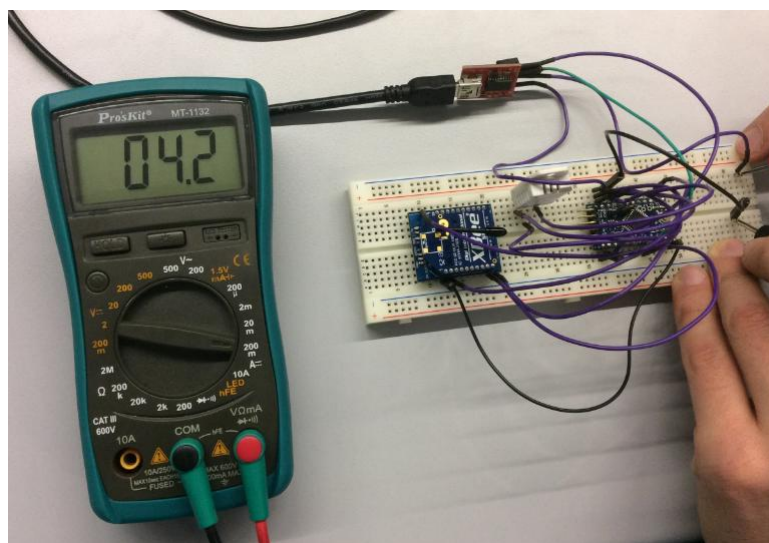


8. ENERGIA KONTSUMOA

Zirkuituan energia kontsumoa murriztea nahi da. Kanpo behatokiko arduinoak, bateria bateri konektatuta dago. Bateriak ez du tentsio handirik emango eta, ahalik eta gehien irauteko, konponenteak lokartzeko programatuko dira. Konponenteak lokartuz, lortuko den helburua izango da, zirkuituaren energia kontsumoa murrizten dela. Energia kontsumoa murrizteko kanpo behatokian, lehenik arduino ProMini-a lokartzeko prestatu egin behar da. Lokartzeko prest programatuta dagoenean, XBee-a lo moduan konfiguratu egingo da ere. Horrela, kanpo behatokiko energia kontsumoa minimizatzea lortu egingo da.

8.1. Arduino promini-aren energia kontsumoa

Kanpoko behatokiko arduinoak eskeintzen duen aukera bat da, arduinoa lokartu egin daitekela eta horrela beraz, behar denean bakarrik piztuta egongo da. Horrerik eskertzen dena da, lokartuta dagoen momentuan energia kontsumoa gutxitzen da, korrontea gutxitzen delako. Arduinoak aurretik duen energia oso handia da. Horretarako arduinoa lokartzeko prestatu behar da. Nahiz eta, 39.irudian ikusten den bezala, arduino ProMini-a ez duen korrante handirik kontsumitzen, korrante kontsumo hori minimizatzea izango da puntu honen helburua.



39.irudia: Arduino ProMini-ren BEREZKO korrante kontsumoa mA-tan



8.1.1. ARDUINO PROMINI LOKARTZEKO PROGRAMAZIOA

1. Lehenik programa nagusiaren "setup" atalean, MCU erregistroko Watchdog Timer (WDT) egoera garbitu behar da

Register Description

MCUSR – MCU Status Register

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
0x34 (0x34)	-	-	-	-	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0					See Bit Description

40.irudia: Arduino ProMini datasheet (54. orrialdea, 11.9.1. atala).

```
void setup() {  
  // Clear the WDT flag in the MCU Status Register. MCU erregistroan WDT egoera garbitu:  
  MCUSR &= ~(_BV(WDRF));  
  
  // put your setup code here, it will run once:  
  Serial.begin(9600);  
  
  xbee.begin(9600); // Xbee abiarazi  
  setupXbee();  
  
  sentsorea.begin(); // HTU21D abiarazi  
  dht.begin(); // DHT abiarazi  
  
  Serial.println("set up Arduino ProMini ondo"); // Monitor Serie pantailan inprimatzeko  
  setupWDT();  
}
```

41.irudia: Arduino ProMini programa.

- 2.- MCU erregistroko WDT garbituta dagoenean, setupWDT() funtzioari deituko zaio, arduinoa lokartze konfigurazioarekin hasteko.

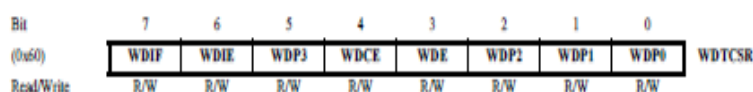
```
void setupWDT() {  
  // ATmega328p manual  
  // Setup the WatchDog Timer (WDT) as indicated in section 11.8  
  // The WDT Control Register (WDTCR) is described in section 11.9.2  
  // Alterations to the Watchdog set-up must follow timed sequences:  
  // 1.- In the same operation, write a logical 1 to the Watchdog change enable bit (WDCE) and  
  WDTCR |= (_BV(WDCE) | _BV(WDE));  
  
  // 2.- Within the next four clock cycles, write the WDE and Watchdog prescaler bits (WDP)  
  // but with the WDCE bit cleared. This must be done in one operation.  
  WDTCR = (1 << WDP2 | 1 << WDP1); // Table 11-2. Watchdog Timer Prescale Select  
  // WDP3 WDP2 WDP1 WDPO Number of WDT Oscillator Cycles Typical Time-out at VCC = 5.0V  
  // 0 1 1 0 128K (131072) cycles 1.0s  
  
  // Table 11-1. Watchdog Timer Configuration  
  // WDTON WDE WDIE Mode Action on Time-out  
  // 0 0 1 Interrupt The "WDT_vect" ISR (Interrupt Service Routine) will be called  
  WDTCR |= _BV(WDIE);  
  
  // Table 10-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes.  
  set_sleep_mode(SLEEP_MODE_PWR_DOWN); // maximum power save
```

42.irudia: Arduino ProMini programa.



3.- Arduino ProMini honen ATmega328p manuala irakurtzen denean WDT-ari buruzko informazioa 11.8 sekziotik aurrera aurkitzen da. 11.9.2 atalean esaten du WDT kontrol erregistroan (WDTCSR) watchdog denborak konfiguratu egin behar direla. Watchdog-eko WDCE eta WDE bit-etan bat logikoa jarri behar dela dio. Aurretik erregistroren baten bat, baldin badago idatzita eta ez ezabatzeko, OR ate logikoa erabiliko da.

11.9.2 WDTCSR – Watchdog Timer Control Register



43.irudia: Arduino ProMini datasheet(54. orrialdea, 11.9.2. atala).

4.- Hurrengo lau erloju zikloetan WDE eta WDP preskaler bit-ak aukeratu nahi diren bezala idatziko dira, hori bai, WDCE bit-a garbiturik egonez. Aukeratu den zikloa segundu batekoa izan da, beraz WDP2 eta WDP1 bat logikoa jarriko zaie eta nola WDCE garbiturik egon behar denez AND ate logiko batekin egiten da erregistroa lehengo kontuan ez hartzeko.

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at $V_{CC} = 5.0V$
0	0	0	0	2K (2048) cycles	16ms
0	0	0	1	4K (4096) cycles	32ms
0	0	1	0	8K (8192) cycles	64ms
0	0	1	1	16K (16384) cycles	0.125 s
0	1	0	0	32K (32768) cycles	0.25 s
0	1	0	1	64K (65536) cycles	0.5 s
0	1	1	0	128K (131072) cycles	1.0 s
0	1	1	1	256K (262144) cycles	2.0 s
1	0	0	0	512K (524288) cycles	4.0 s
1	0	0	1	1024K (1048576) cycles	8.0 s

44.irudia: Arduino ProMini datasheet (55. orrialdea, 11.9.2. atala).



5.- Watchdog Timerra konfiguratzeko momentua heldu da. Aukeratzen da beharrezkoa den modua, kasu honetan Interrupt Mode-a aukeratu da. WDIE bit-ean bat logikoa jarriko zaio eta aurretik zeuden erregistroko datuak kontuan artzeko OR ate logikoarekin jarriko da. WDT denbora bukatzean "WDT_vect" ISR-ri deituko dio.

```
Arduino IDE: xbee_proba_kanpoko1 Arduino 1.6.12
Archivo  Editar  Programa  Herramientas  Ayuda
xbee_proba_kanpoko1 $
,
// Watchdog Interrupt Service.
// This function is executed when watchdog times out.
ISR(WDT_vect) {
  // arduinoa 20 segunduz lokartu
  if (wdt_count < 20)
    wdt_count++;
  else {
    f_wakeup = 1;
    wdt_count = 1;
  }
}
```

45.irudia: Arduino ProMini programa.

ISR interrupzioa heltzean programa agindu hauek beteko ditu. Lehen adierazi den bezala, lo zikloak segundu batekoak dira eta arduinoa 20 segunduz lokartuta egotea nahi denez, if funtzio batean kontadore bat jartzen da segunduen kontaketa egiteko. Kontadorea $wdt_count < 20$ baino txikiagoa baldin bada, bat gehituko dio eta berriz ere konparatzen segituko du 20ra heldu arte. Orduan, wdt_count 20ra heltzen denean, *else* funtziora pasatuko da eta wdt_count kontagailua eta f_wakeup parametroei bat logikoa jarriko zaie.

WDTON ¹⁾	WDE	WDIE	Mode	Action on Time-out
1	0	0	Stopped	None
1	0	1	Interrupt Mode	Interrupt
1	1	0	System Reset Mode	Reset
1	1	1	Interrupt and System Reset Mode	Interrupt, then go to System Reset Mode
0	x	x	System Reset Mode	Reset

Note: 1. WDTON Fuse set to "0" means programmed and "1" means unprogrammed.

46.irudia: Arduino ProMini datasheets (55. orrialdea, 11.9.2. atala).



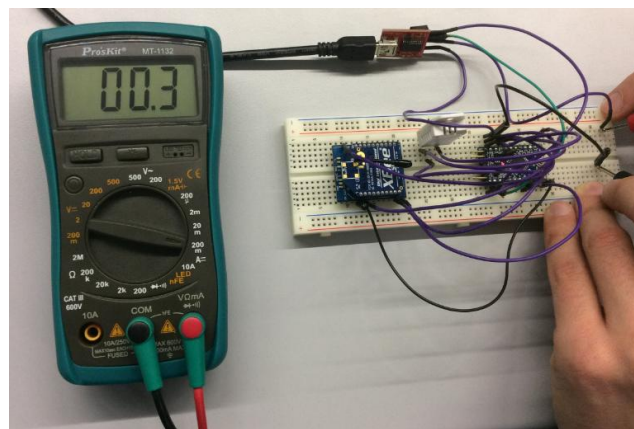
6.- Lotara joateko aktibazioa eta esnatzeko aktibazioa, modulua lo geratzeko modua aukeratu.

Programa nagusian enterSleep funtzioari deitzean, bere azpi-programara salto egiten du egin beharreko aginduak betetzeko. Azpi-programan sartu eta egin beharreko lehen agindua da, gailu guztiak desaktibatu. Gero, f_wakeup parametroa 0 den bitartean lo moduan sartuko da, CPU lokartuz eta ondoren lo modutik desgaituz. Baina f_wakeup = 1 baldin bada, ez da *while*-ko lo moduan sartuko arduinoa eta egingo duena izango da f_wakeup parametroa 0an jarriko du berriz ere eta gailu guztiak berriz aktibatuko ditu.

```
xbee_proba_kanpoko1 Arduino 1.6.12
Archivo Editar Programa Herramientas Ayuda
xbee_proba_kanpoko1 $
,
// Watchdog Interrupt Service.
// This function is executed when watchdog times out.
ISR(WDT_vect) {
  // arduinoa 20 segunduz lokartu
  if (wdt_count < 20)
    wdt_count++;
  else {
    f_wakeup = 1;
    wdt_count = 1;
  }
}
```

47.irudia: Kanpo behatokiko arduino programa.

Hurrengoko irudian ikus daitekeen bezala, arduino ProMini-aren korrontea dezente jeizten da. Bere hasierako korrontearekin alderatuz %92a jeizten da, beraz, energia kontsumoa asko minimizatu egingo da.



48.irudia: Arduino ProMini-aren korronte kontsumoa LO dagoenean (POWER_DOWN lo modua).



8.2. XBee-aren energia kontsumoa

Kanpo behatoki honetan, XBee-aren energia kontsumoa txikitu daiteke ere. XBee-ak atseden moduan programatu daitezke erabiltzen ez direnean. Atseden modu horietan sartzean, kontsumo baxuko egoeran sartzen dira. Atseden egoeran sartzeko badintza batzuk bete behar dira (SM zero ez den beste egoera batean):

- Sleep_RQ (pin9) konfirmatzen da eta modulua pinaren itxaron modulua (SM = 1, 2 edo 5).
- Modulua ez dago aktibaturik definitutako ST denboran (ST bakarrik dago aktibo SM 4 edo 5).

XBee-ak atseden moduan sartzeko bost modu ezberdin daude:

- **Pin Hibernate → SM = 1**

Hibernate Mode pin honek potentzia kontsumoa minimizatzen du ez dagoenean aktibo Xbee-a. Sleep_RQ (pin 9) agintzen denean, modulua edozein agindu bukarazten du eta Xbee modulua inaktibo egoeran sartzen da eta lo moduan sartzen da. Modulua ez du erantzunik emango inaktibo egoeran sartzean. Lo dagoen modulua itzartzeko eta hibernate moduan lan egiteko, Sleep_RQ (pin 9) desaktibatu behar da. Hau da, 0an dagoenean pin-a XBee-a esnaturik egongo da eta 1ean dagoenean, aldiz, lo egongo da .

- **Pin Doze → SM = 2**

Pin Doze Mode-a, iratzarpen denbora bizkorra du, 2ms-koa eta energia kontsumo handiagoa eskeintzen du Hibernate Mode-arekin konparatuz. Pin Doze Mode-an atseden modua aktibatzeke, Sleep_RQ (pin 9) desaktibatu behar da. Aktibatu egingo da Sleep_RQ desaktibatua dagoenean eta prest dagoenean transmititzeko eta jasotzeko, CTS lerroa behean egongo da. Modulua itzartzean, pin-a "de-afirmarlo" bi bytes aldiz, CTS-a eta gero. Horrela denbora izango du datuak bufferrean sartzeko.



- Lo ziklo Modua → SM = 4

Zikloka atsedean hartzen duten moduluak periodoka konprobatzen dute irrati frekuentziatik datozen datuak. SM parametroa 4 zenbakian konfiguratu denean lo egiteko prest dago moduluak, zikloka iratzartzen da. Moduluak konfiguratu da "coordinador Calculo del Sueño" bezala (SM = 0, CE = 1). The cyclic Sleep Remote bidaltzen dio koordinadoreari ikerketa bat egiteko SP (lo ziklo periodoa) parametroak ezarritako tarteen artean. Koordinadorea datu guztiak "kolara" bidaliko ditu. "Kolara" ez badaude daturik lo zikloan geratuko da. "Kolan" datuak baldin badaude komunikatu egingo dira datuak ST (lo egin aurreko denbora) tenporizadorea esan arte. Urruneko kontrola aktibatzen den bakoitzean CTS desaktibatu egiten da horrela, hasierako komunikazioa lagaz.

- Urruneko kontroleko lo zikloa Wake-up Pin-arekin (SM = 5)

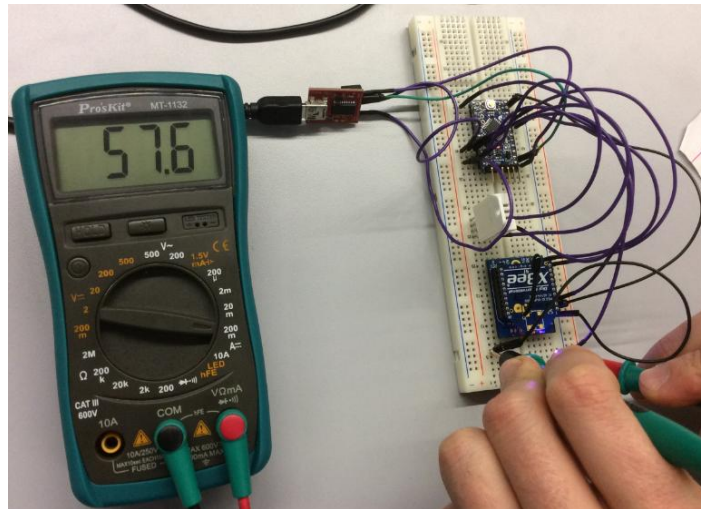
Modu hau erabiltzen da, urruneko kontroleko modulu bat atsedean egoeran izteko irrati frekuentziaren bidez edo Sleep_RQ deskribatuz pultsuen bitartez komunikatzeko. Atsedean ziklo modua iratzargailu batekin kontrolatzen da urruneko moduluko pin-etik. Moduluak aktibatu egingo da detektatzen duenean, eta orduan CTS baxura ajustatuko da transmititu edo jasotzeko.

- Lo ziklo koordinadorea (SM = 6)

Modu honekin irrati frekuentziaren interkonexioaren bidez urruneko kontroleko lo zikloak kontrola daiteke. Koordinadoreak urrutiko helbidetik datorren mezua onartu egingo du eta bufferrean gorde egingo da itxartu arte eta beste agindu bat izan arte.



XBee-ak ez daudenean energia aurrezteko moduan konfiguratutik, kontsumitzen duten korrontea hurrengokoa da:



49.irudia: XBee-aren BEREZKO korronte kontsumoa mA-tan.

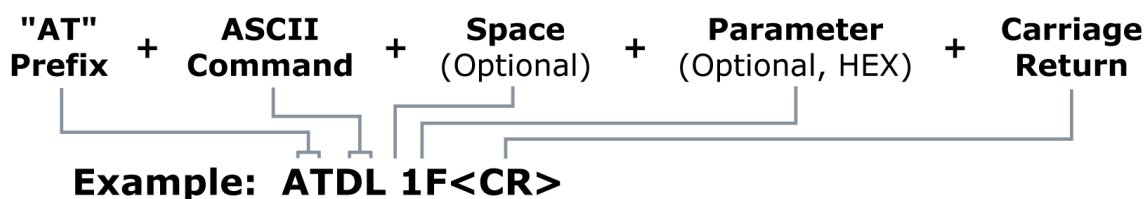
8.2.1. XBee-aren lokartzeko programazioa

XBee-aren lokartzeko pin-a eta arduinoaren arteko konexioa, kable baten bidez egiten da; non XBee adapter-eko 9.pin-etik, arduinoko 6.pin-erarte doana. XBee-an lokartzeko moduan sartzeko *setupXbee* funtzioari deituko zaio. Arduinoko 6.pina irteera bezala defintu egingo da eta desaktibatuta utziko da(lo modua desaktibatuta arduino eta XBee-aren artean). XBee-a abiarazi egiten da. *AT komandoz* moduan idazteko, XBee-aren datasheet-eko *AT komandoz* atalean dauden aginduak idatzi behar dira XBee-ari, komandoz moduan sartzeko.

- *AT komandoz* aginduak sartu baino lehenago segundu bateko denbora tartea utzi behar zaio XBee-a ondo abiarazteko.
- Denbora hori pasatzean XBee-ari “+++” karaktereak bidaliko zaizkio *AT komandoz*-ean sartzeko.



- XBee-ari denbora tarte bat eman beharko zaio *AT komandoz*-en sartzeko eta "OK" batekin erantzun bat jasotzeko. "OK" erantzun hori jasotzean, Xbee-a *AT comandos* egoeran dagoela esan nahiko du.
- "OK" hori heldu dela konprobatzeko, *while* baten barruan galdetzen zaio ea xbee-ak zeozer jaso duen, agindu honekin: `xbee.available()>0`. Xbee-an karaktere baten bat jaso baldin badu Monitor Seria-n inprimatzeko esango zaio. Erantzuna jasotzen denean jakingri da *AT komandoz-en barruan dagoela*.
- Hurrengo aginduak bidaltzeko erabili beharreko sintaxia honako da:



50.irudia: *XBee-aren datasheet* (25. orrialdea, "command mode" atalean).

- XBee-arekin lortu nahi den helburua nagusia, energia kontsumoa minimizatzea da. XBee-k lokartze modu ezberdinak aukeratzeko dituzte baina energia kontsumo gutxien eskeintzen duena SM=1 modua da. XBee-aren energia kontsumoa 10 μ A baino gutxiagora jeisten da.

Sleep Mode Setting	Transition into Sleep Mode	Transition out of Sleep Mode (wake)	Characteristics	Related Commands	Power Consumption
Pin Hibernate (SM = 1)	Assert (high) Sleep_RQ (pin 9)	De-assert (low) Sleep_RQ	Pin/Host-controlled / NonBeacon systems only / Lowest Power	(SM)	< 10 μ A (@3.0 VCC)
Pin Doze (SM = 2)	Assert (high) Sleep_RQ (pin 9)	De-assert (low) Sleep_RQ	Pin/Host-controlled / NonBeacon systems only / Fastest wake-up	(SM)	< 50 μ A
Cyclic Sleep (SM = 4)	Automatic transition to Sleep Mode as defined by the SM (Sleep Mode) and ST (Time before Sleep) parameters.	Transition occurs after the cyclic sleep time interval elapses. The time interval is defined by the SP (Cyclic Sleep Period) parameter.	RF module wakes in pre-determined time intervals to detect if RF data is present / When SM = 5	(SM), SP, ST	< 50 μ A when sleeping
Cyclic Sleep (SM = 5)	Automatic transition to Sleep Mode as defined by the SM (Sleep Mode) and ST (Time before Sleep) parameters or on a falling edge transition of the SLEEP_RQ pin.	Transition occurs after the cyclic sleep time interval elapses. The time interval is defined by the SP (Cyclic Sleep Period) parameter.	RF module wakes in pre-determined time intervals to detect if RF data is present. Module also wakes on a falling edge of SLEEP_RQ	(SM), SP, ST	< 50 μ A when sleeping

51.irudia: *XBee-aren datasheet* (23. orrialdea, "Sleep mode" atalean).



- Aukeratu den modua agintzeko, 51.irudian jarritako sintaxi eredua erabili behar da aginduak ezartzeko. XBee-ari aukeratutako modua idazteko horrela idazten zaio, `xbee.print("ATSM1")` eta ondoren ASCII kodea erabiliz XBee-ari orga-itzulera(CR) agindua idazteko agintzen zaio, `xbee.write(0x0D)`. Hasierako agindua bidali eta gero egin den bezala, denbora tarte bat utzi beharko zaio erantzuna jasotzeko.

Oharra: plataforma ezberdinetan "enter" era ezberdinetan inplementatzen da:

- Windows: `\r\n` = CR LF
- Linux: `\n`
- Mac: `\r`
- XBee: `\r`
- `println` → <https://www.arduino.cc/en/serial/println> → `ln = \r\n` = CR LF = `0x0D 0x0A`

Kasu honetan, sintaxi irudia deskribatuta dagoen bezala, aginduaren ondoren CR agindua sartu behar zaio eta ez `println` hasieran egin den bezala, zeren eta, `println CR eta LF` aginduak ditu eta plataforma honetan ez ditu behar bi agindu horiek, CR agindua bakarrik baizik. Horregaitik ASCII kodearen bidez idazten zaio XBee-ari CR agindua. `CR = 0x0D` → `xbee.write(0x0D)`.

- Lehen erabilitako konprobazio bera erabiltzen da, ikusteko ea XBee-ak zeozer idatzi duten eta erantzuna jasotzen den.
- XBee-a *AT komandoz-etik* irtetzeko, "ATCN" agindua idatzi behar zaio eta orga-itzulezko ASCII kodea ere. Horrela idatziko dira aginduak. `xbee.print("ATCN"); xbee.write(0x0D);`
- Aurretik erabilitako denbora tarte eta konprobazio bera egiten da, bidalitako agindua heldu den ikusteko, erantzuna jasotzeko eta *AT comandos-etik* irten dela jakinarazteko.



```
xbee_proba_kanpoko1 Arduino 1.6.12
Archivo  Editar  Programa  Herramientas  Ayuda

xbee_proba_kanpoko1 $
void setupXbee() {
  pinMode(6, OUTPUT);
  digitalWrite(6, LOW);
  xbee.begin(9600);
  delay(1000);

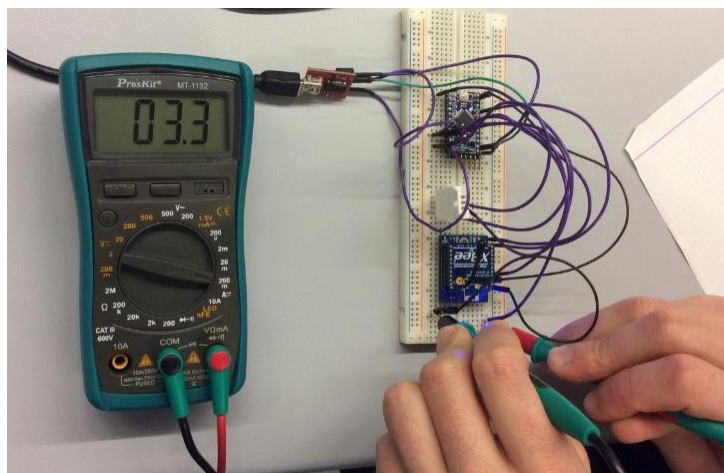
  xbee.print("+++");
  Serial.println("+++");
  delay(10000);
  while(xbee.available()>0) {
    Serial.print(char(xbee.read()));
  }
  Serial.println();

  xbee.print("ATSM1"); xbee.write(0x0D);
  Serial.println("ATSM1");
  delay(3000);
  while(xbee.available()>0) {
    Serial.print(char(xbee.read()));
  }
  Serial.println();

  xbee.print("ATCN"); xbee.write(0x0D);
  Serial.println("ATCN");
  delay(3000);
  while(xbee.available()>0) {
    Serial.print(char(xbee.read()));
  }
}
```

52.irudia: XBee-aren arduino programa.

Kasu honetan ere ikus daiteke, XBee-aren korrante kontsumoa nabarmenkin minimizatu egin dela. XBee honek izan duen korrante jeitsiera %94koa izan da, ondorioz minizatu den energia kontsumoa oso handia da.

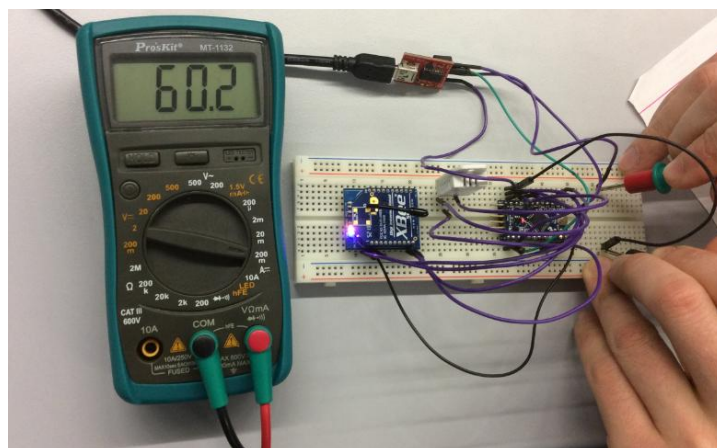


53.irudia: XBee-aren korrante kontsumoa mA-tan, LO dagoenean (Hibernate lo modua).

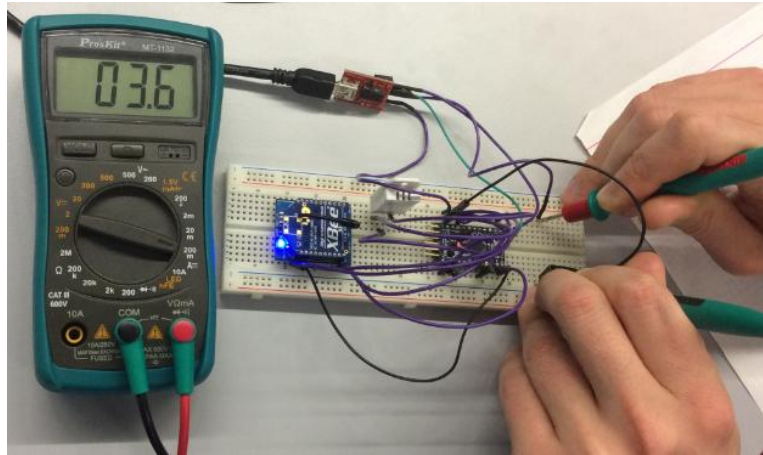


Komandoz modu hau, arduino programan programatzen da. Arduino programan komandoz modua idaztean, XBee-ak bakarrik programa hau kargatzean sartuko da komandoz moduan. Beste aukera bat da, X-CTU programan XBee-ari hibernate modua jartzea, baina honekin lortzen dena da, XBee-a beti hibernate modura programatuta egongo dela. Orduan erabaki egin da, arduino programan XBee-ari komandoz moduan konfiguratzea, bakarrik programa horretan hibernate moduan sartzeko.

Bi elementuak (Arduino eta XBee-a) batera funtzionatzen daudenean kontsumitu egiten duten korrontearen batazbesteko 60.2mA-koa da, 54.irudian ikus daiteken bezala. Aldiz, Arduino eta XBee lotara bidaltzen direnean korronte kontsumoa 3.6mA-tara murriztu egiten da, 55.irudian ikusten den bezala. Hortaz, bi konponente nagusi hauetan aplikatutako kontsumo aurrezpen prozedurak, %94 kontsumo aurrezpena dakar.



54.irudia:Arduinoa eta XBee-aren arteko BEREZKO korronte kontsumoa mA-tan.



55.irudia: Arduinoa eta XBee-aren arteko korrante kontsumoa mA-tan, LO dagoenean.



9. DATUAK HODEIRA IGO

Kanpo eta barne behatokiek hartutako datuak edo laginak hodeian gordetzeko, ThingSpeak.com datu biltegi eta datu bistaratzailerak aukeratu da. Biltegi honek REST API bat dauka (hau da, HTTP protokoloan oinarritutako API-a), API horretan definituriko ekintzen arabera biltegia programatikoki kudeatu daitekeelarik. REST API-a ondorengo estekan definitzen da: <https://es.mathworks.com/help/thingspeak/channels-and-charts.html>

ThingSpeak.com biltegia kanaletan antolatzen da, kanal bakoitza 8 datuz edo eremuz osoturik egon daitekeelarik.

Name	Created
Logger Private Public Settings API Keys Data Import / Export	2016-11-28
Ordenador Private Public Settings API Keys Data Import / Export	2017-01-12
Nire kanala Private Public Settings API Keys Data Import / Export	2017-02-07

56.irudia: ThingSpeak orria.

Kanala sortzeko inprimaki bat bete behar da, bertan kanala osotuko duten datuak (kasu honetan, tenpB, hezeB, tenpK eta hezeK) definitzendirelarik.



Channel Settings

Percentage complete 50%

Channel ID 224139

Name Nire kanala

Description Eguraldi Behatokia

Field 1 tenpB

Field 2 hezeB

Field 3 tenpK

Field 4 hezek

57.irudia: ThingSpeak web-guneko datuak.

Datuak igotzeko API-ak definituriko “Update a Channel Feed” ekintza erabili behar da.

Update a Channel Feed

To update a channel feed, send an HTTP GET or POST to <https://api.thingspeak.com/update>.

Valid parameters:

- **api_key** (string) - Write API key for this specific channel (required). The Write API key can optionally be sent via a THINGSPEAKAPIKEY HTTP header.
- **field1** (string) - Field 1 data (optional)
- **field2** (string) - Field 2 data (optional)
- **field3** (string) - Field 3 data (optional)
- **field4** (string) - Field 4 data (optional)
- **field5** (string) - Field 5 data (optional)
- **field6** (string) - Field 6 data (optional)
- **field7** (string) - Field 7 data (optional)
- **field8** (string) - Field 8 data (optional)

58.irudia: Aginduak idazteko modua.



API-ak adierazten duenaren arabera, ekintza hau burutzeko erabili behar den HTTP eskaerak ondorengo parametroak izan behar ditu:

- Zerbitzaria: api.thingspeak.com
- Portua: 443 (https)
- Metodoa: GET edo POST
- URI-a: /update
- Inprimaki formatuko datuak:
 - api_key: kanalean idazteko gakoa
 - fieldX: igo nahi diren datuak

HTTPS eskaera bat ezin daiteke Arduino burutu, ez baitu HTTPS-ek beharrezko dituen SSL edo TTL protokoloen zifratze algoritmoak prozesatzeko ahalmenik. Horregaitik erabiltzen dugu Arduino Yun-a, hain zuzen ere, honek Arduino mikrokontroladoreaz aparte, Linino Linux banaketa duen mikroprozeadorea daukalako, eta azken honek badu HTTPS kudeatzeko gaitasuna. Horrez gain, Arduino Yun-ek badu baita ere HTTP eskaerak egiteko beharrezkoa den sareko interfazea, WiFi interfazea, alegia.

HTTP eskaerak programazio lengoia askotan programatu daitezke. Programazio lengoia guztien artean, GRaL honetako zuzendariak Python erabilpena gomendatu zuen, HTTP eskaerak era sinplean eta argian egiteko aukera ematen baitu. Bestetik, Linux banaketa gehienek Python interpretea berez dakarre.

Hortaz, Python-en ThingSpeak-com -era 4 daturen bidalketa egiten duen script-a programatu da (thingspeak3.py) aurreko argibideak kontutan izanik.

Zelan deitzen da Arduinotik Linux aldean exekutatu behar den Python script-a? Lehenengo eta behin argi izan behar da Linux-eko kontsolan Python script-a zelan exekutatu genukeen:

```
python /root/thingspeak3.py 22.3 76.5 30.1 55.2
```



Arduino Yun-en Arduino aldean, beste Arduino batzutan erabili ezin daitekeen liburutegi bat erabili daiteke, Arduino eta Linino konektatu egiten dituen: Process.h. Horrela, Arduinoa eta Linux-en arteko elkar konexio iturria abiarazi egiten da *Bridge.begin()*; agindua idatziz. Datuak hodeira igotzeko *uploadData2Cloud* funtzioa definitu da, bertan bidaliko diren parámetro bezela izendatzen direlarik.

```
void uploadData2Cloud(float hezeK, float tenpK, float hezeB, float tenpB) {
    Process prozesua;
    prozesua.begin("python");
    prozesua.addParameter("/root/thingspeak3.py");
    prozesua.addParameter(String(hezeK));
    prozesua.addParameter(String(tenpK));
    prozesua.addParameter(String(hezeB));
    prozesua.addParameter(String(tenpB));
    prozesua.runAsynchronously();
    unsigned long start = millis();
    while(prozesua.running()) {
        unsigned long now = millis();
        if((now - start) > 3000)
            prozesua.close();
        else
            delay(100);
    }
}
```

59.irudia:Arduino programan programatutakoa.

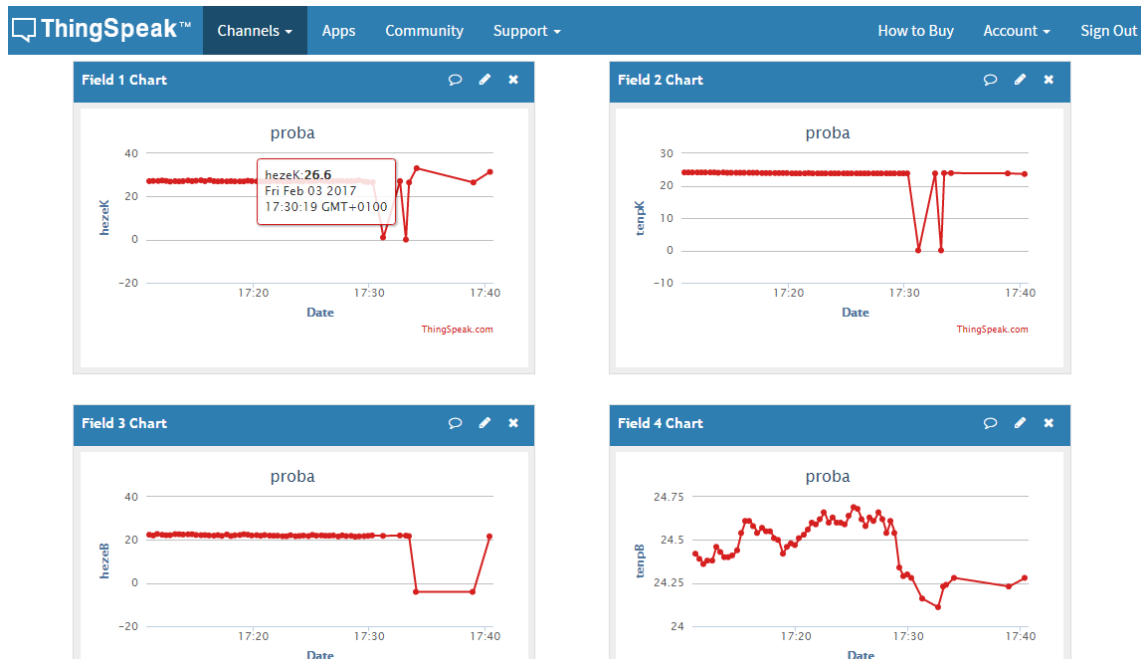
Kodearen azalpena.

- Funtzioan, parametroak definitu egingo dira: *float hezeK*, *float tenpK*, *float hezeB*, *float tenpB*.
- Process, prozesua izenarekin definitzen da eta Linux aldean *python* programa abiarazteko agintzen zaio.
- */root/thingspeak3.py* izeneko parametroa definitzen da prozesuan.
- Ondoren, bidali beharreko parametro guztiak adierazten dira.
- Prozesua martxan jartzeko agintzen zaio, *prozesua.runAsynchronously()*;
- Prozesua hasten deneko uneanerutue egiten da: *unsigned long start = millis()*;



- Prozesua Linux aldean martxan dagoen bitartean *while*-aren barnean sartu egingo da eta berriz ere denbora artuko du.
- *if*-aren barnean sartzeko bete beharrekoa ondorengoa da: prozesua abiarazi zenetik hona zenbat denbora pasatu den neurtu; denbora hori 3000 (3s) baino handiago bada, prozesua hil egiten da; aldiz, prozesuak 3s baino gutxiago badaramatza exekututzen, orduan Arduinoko aldeak veste 100 milisegunduz itxarongo dio eta berriz ere *while*-era itzuliko da prozesua martxan dagoela konprobatzeko. Horrela Linux aldeko prozesua blokeatu al den egiaztatzen da eta, blokeatu baldin bada, Arduino aldeak prozesua hiltzeko agindua ematen du. Beste ikuspuntu batetik begiratuta, Arduino kode zati hau gehienez ere 3 segunduz exekutatu da, 3s horien ondoren tenperatura eta hezetasun lagin berriak irakurriko dituelarik.

Sareko konexioak hutsegiterik ez badu, datuak ThingSpeak-ko kanalera igo eta bertan bistaratuko dira.



60.irudia: ThingSpeak-en behatokitako datuak marraztu grafikoan.



Datu horiek ThingSpeak-en daudelarik, edozein erabiltzailek MATLAB-etik jeitsi ditzake ThingSpeak toolbox-a erabiliz:

<https://es.mathworks.com/matlabcentral/fileexchange/52244-thingspeak-support-toolbox>.

MATLAB eta ThingSpeak.com -en arteko integrazioa bermaturik dago MATLAB ThingSpeak zerbitzuaren jabe delako.

Ondorengo irudian, ThingSpeak toolbox-eko thingSpeakRead() funtzioa erabiliz kanaleko datuak irakurri eta MATLAB-era jeisten dira.

```
>> data = thingSpeakRead(219541, 'Fields', [1,2,3,4], 'NumPoints', 30, 'Timeout', 10, 'OutputFormat', 'table')
```

```
data =
```

Timestamps	hezeK	tenpK	hezeB	tenpB
03-Feb-2017 17:22:11	27	23.8	21.95	24.62
03-Feb-2017 17:22:33	27	23.8	21.72	24.66
03-Feb-2017 17:22:55	27.1	23.8	21.71	24.6
03-Feb-2017 17:23:17	26.9	23.8	22.18	24.63
03-Feb-2017 17:23:39	27.2	23.8	21.73	24.6
03-Feb-2017 17:24:02	27	23.8	21.85	24.6
03-Feb-2017 17:24:24	27	23.8	22.02	24.59
03-Feb-2017 17:24:46	27.2	23.8	21.79	24.64
03-Feb-2017 17:25:09	27.1	23.8	22.34	24.69
03-Feb-2017 17:25:30	27.2	23.8	21.97	24.68
03-Feb-2017 17:25:53	27.4	23.8	22.09	24.62
03-Feb-2017 17:26:15	27.2	23.8	21.94	24.58
03-Feb-2017 17:26:37	27.1	23.8	21.93	24.63
03-Feb-2017 17:26:59	27.1	23.8	22.08	24.61
03-Feb-2017 17:27:22	27.2	23.8	21.6	24.66
03-Feb-2017 17:27:44	27.2	23.8	22.1	24.62
03-Feb-2017 17:28:06	27.1	23.8	21.79	24.54
03-Feb-2017 17:28:28	27.1	23.8	21.89	24.61
03-Feb-2017 17:28:50	27.2	23.8	21.5	24.54
03-Feb-2017 17:29:13	27.5	23.8	21.66	24.34
03-Feb-2017 17:29:35	26.9	23.8	21.76	24.29
03-Feb-2017 17:29:57	26.6	23.8	21.9	24.3

61.irudia: Matlab-en datuak aurkeztuak.

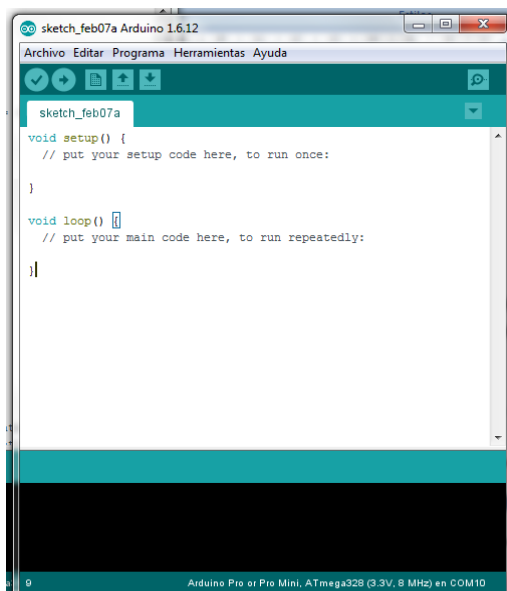


10. SOFTWARE PROGRAMAK

Proiektu hau burutzeko, software programa ezberdinak erabili dira. Programazioaren kasuan, arduino programa erabili da. Zirkuituen konexioak irudikatzeko, fritzing programa aukeratu da. Zirkuitu integratu plaka diseinatzeko, designSpark programa erabili da.

10.1. ARDUINO programa

Bi behatokiaren programazioa egiteko, arduino programa erabili da. Programa honek, programatzeko erretasun handiak ematen ditu. Konponente elektroniko ezberdinen programazio liburutegiak, arduino programara gehitu behar dira lehenengo. Liburutegiak programan sartuta daudenean, programa programatzeko hasteko prest egongo litzake. Programa C lengua bitartez idazten dira aginduak.

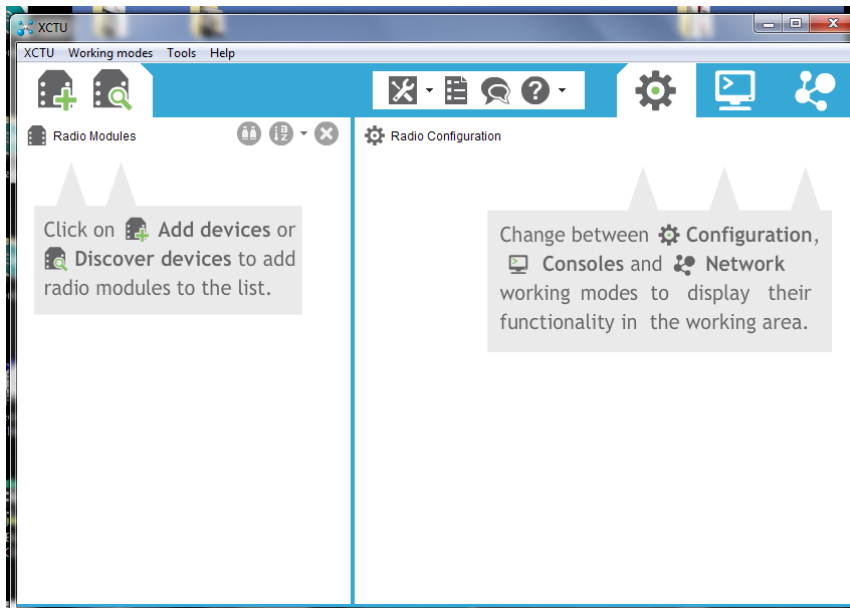


62.irudia: Arduino programa.



10.2. X-CTU programa

Programa hau erabili egin da, irrati frekuentziaren bidez doazen txip-ak elkar konfiguratzeko. Txip hauek, hau da, XBee-ak, bata bestearekin komunikatzeko X-CTU programan konfiguratzen behar dira. Programa honetan, datu espezifiko batzuk sartu behar dira elkar komunikazioa lortzeko.



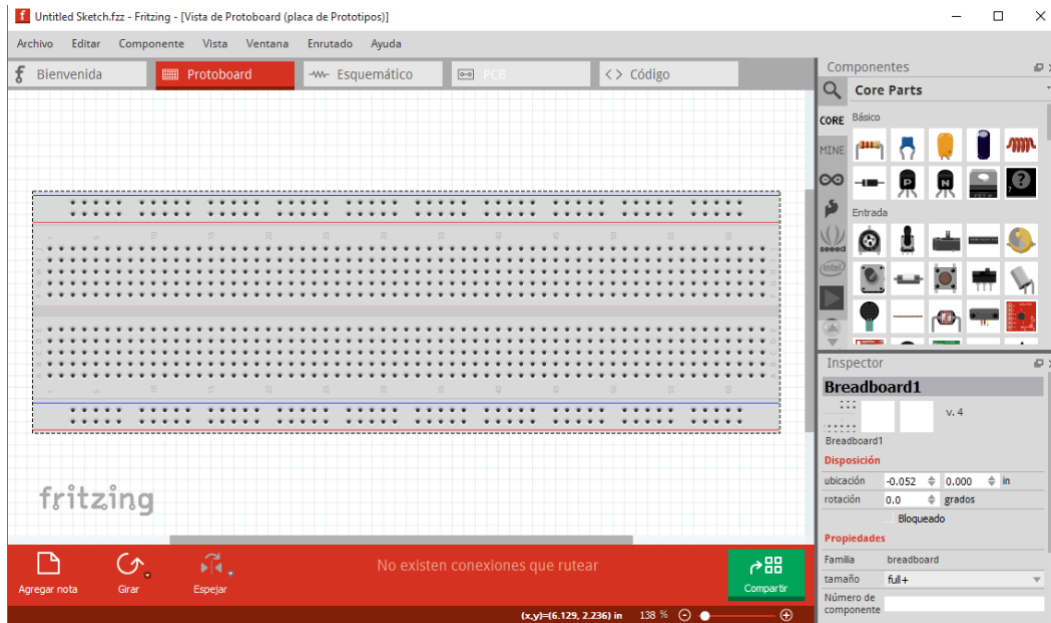
63.irudia: X-CTU programa.

10.3. FRITZING programa

Zirkuituen konexioak argiago ikusteko eta hobeto deskribatzeko fritzing programa erabili da. Programa honetan konponente elektroniko ezberdinak aukeratzeko aukera ematen du. Fritzing programa honek, konponente ezberdineko liburutegiak sartuta ditu eta hortik aukeratzeko beharrezkoak diren konponenteak. Liburutegi horietan agertzen diren konponente elektroniko ezberdinetako irudiak aukeratu daitezke. Konponente elektroniko bat behar baldin bada, eta ez badago programan sartuta; ez dago arazorik, konponente



horren liburutegia internet-en aurkitu esker. Fritzing programak aukera ematen du, aurkitutako liburutegi berri hori bere programako fitxategira igotzeko. Horrela, programako liburutegira joanez, konponentearen fitxategia aurkituko da eta bere irudia aukeratu ahalko da.

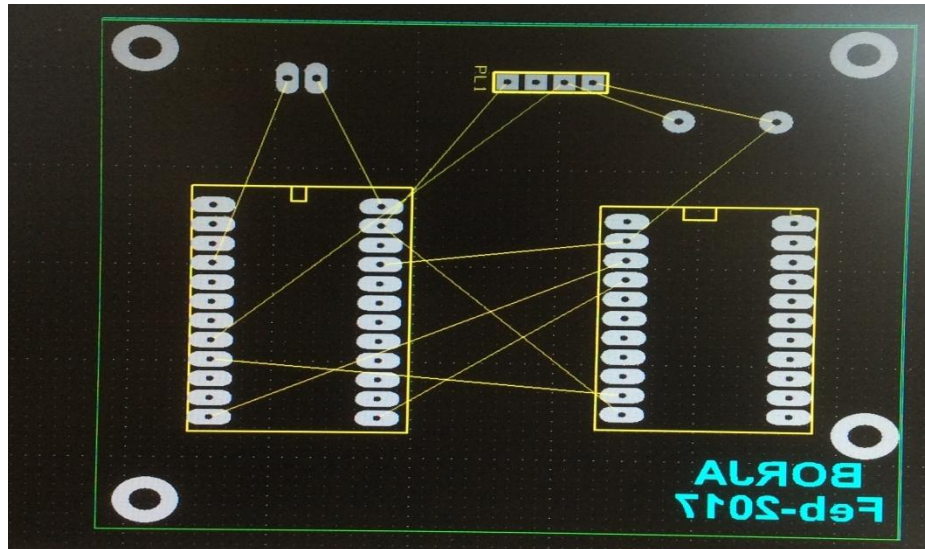


64.irudia: Fritzing programa.

10.4. DESIGNSPARK programa

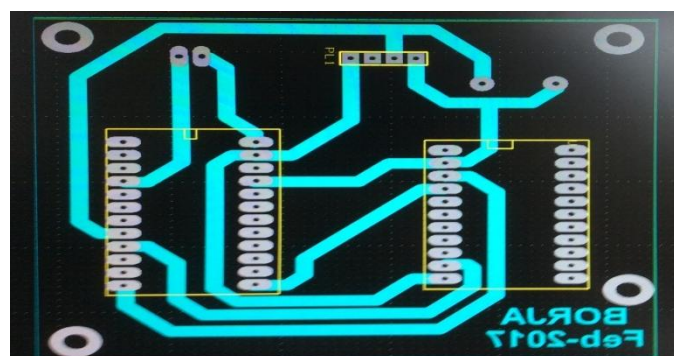
Behatokitako zirkuituak, konponente elektronikoz guztien arteko konexioa kableen bidez egin da. Kable guzti horiek, era txukunago batean agertzeko, zirkuitu integratu batez aldatzeko aukera egon da. Zirkuitu integratuak diseinatzeko erabili den programa, DesignSpark izan da.

Programa honetan, konponente elektronikoz ezberdinak aukeratzeko aukera dago programaren barneko liburutegian. Hori esanda egin beharreko da, aukeratu behar diren konponente elektronikoz guztiak liburutegitik hautatu eta jarri nahi diren posizioetan kokatu dira. Konponente guztiak pantailan daudenean, elkarren arteko konexioa egiten hasiko da.



65.irudia: DesignSpark programako konpone konexioak.

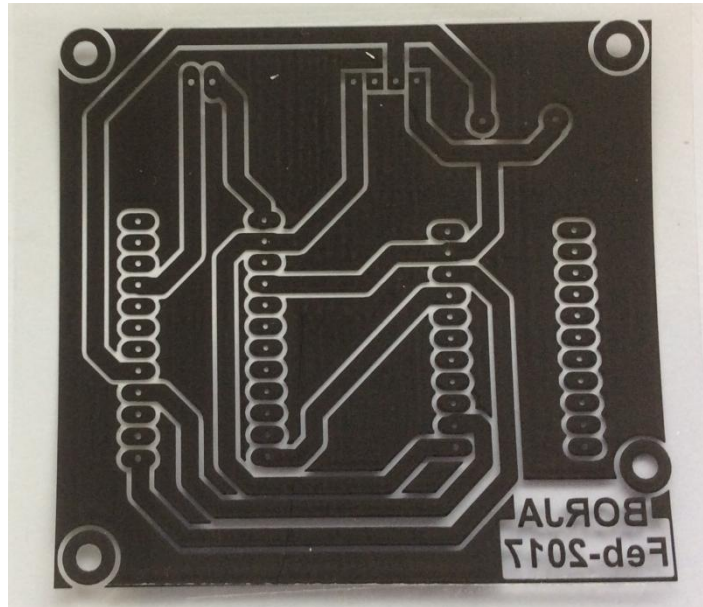
Konponete elektronik guztien arteko konexioak eginda daudenean, DesignSpark programak ematen duen aukeretako bat da, kableen konexioak bera bakarrik ordenatzen dituela, bata bestearekin ez gurutzatzeko. Aukera honeri esker, konexio gehienak eginak egongo dira elkar gurutzatu barik. Egon daiteke konexioren bat egin gabe utzi izana eta horretarako bi aukera daude konexio horrek egiteko; bata, konponete elektronikoak posizioz aldatu eta ea hórrela bide berriak aurkitu ditzakeen programak eta bestea, manualki falta diren kableak konektatu bideak elkar gurutzatu barik.



66.irudia: DesignSpark programan konexio finala.

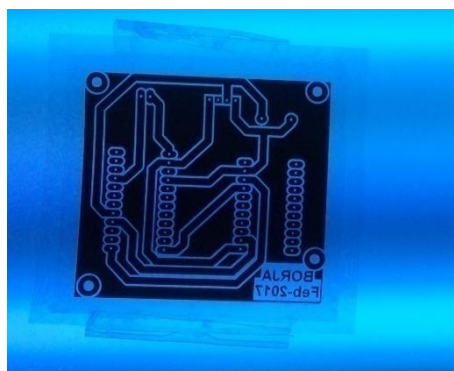


Diseinatutako zirkuitua, konponente elektroniko guztiekin eta egin beharreko konexio guztiekin egina baldin badago, inprimatzea bakarrik falta da. Zirkuitua azetato paperean inprimatzean da.



67.irudia: Zirkuitua azetato papelean inprimatua.

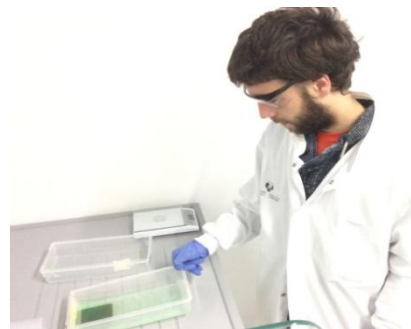
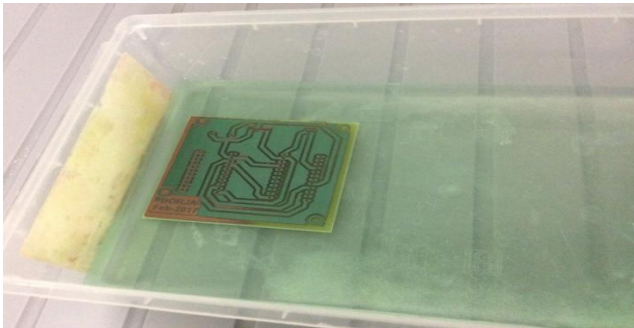
Azetato paperean inprimatu den zirkuitua, kobrezko plaka baten gainean jartzen da eta "insoladore" makinan sartu egiten da, zirkuitua plakan kopiaturi geratzeko. Lau minutu pasatzean "insoladore" makinatik kobrezko plaka ateratzen da eta kableen bideak marraztuta daudela ikusi beharko da.



68.irudia: Kobrezko plaka eta azetato papela, insoladore makinan.

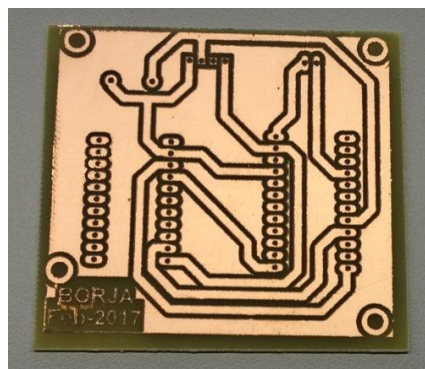


Kobrezko plakaren zirkuitua hobeto ikusteko, errebelatzaile likidoan sartzen da fotolitoak isladatzeko. Denbora pixka bat pasatzean urarekin garbitzen da plaka. Ondoren, atakatzaila den likido batean sartu egingo da plakako kobrea kentzeko. Bideak kobrea kentzen hasten denean erreakzioa egiten ari dela esan nahiko du eta likidotik atera egingo da, uretara pasatuz eta berriz ere garbituz.



69.irudia: Kobrezko plaka atakatzea.

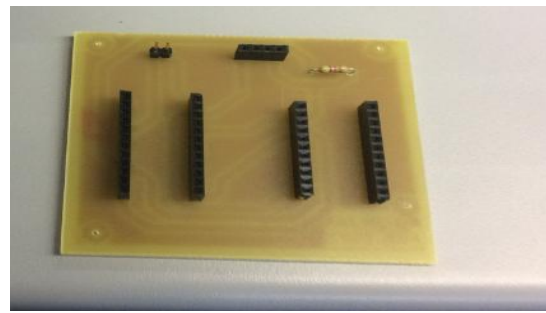
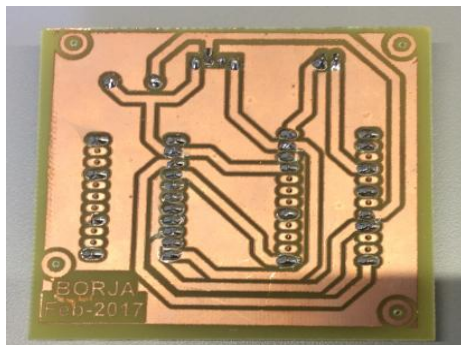
Plaka uretatik garbitzen denean eta lehortzen denean, berriz ere “insoladore” makinan 5 minutuz sartuko da. Denbora hori pasatzean berriz ere likido bereziarekin atakatu egingo zaio bideetako kobrea kentzeko.



70.irudia: Kobrezko plaka garbitu ondoren.



Plakaren zirkuitu integratua ondo ibiltzeko egindako pauso guztien ondoren falta den pausoa hurrengokoa da, konponente elektroniko bakoitza plakan jartzea. Hori egiteko pin gehigarriak jarri behar dira plakan. Irudian ikusten den bezala zirkulu bakoitza, konponente bakoitzaren pin-entzat dira eta zultatzeko makina erabilia, zuloak egingo dira pin gehigarriak jartzeko. Pin gehigarriak, plakari soldatuko zaizkio kontu handiz. Horrela plakan, konponente elektronikoak jartzea bakarrik faltako zen.



71.irudia: Kobrezko plakaren azken emaitza.



12. BIBLIOGRAFIA

- Seidle, N. (2006), Sparkfun HTU21D,
<https://learn.sparkfun.com/tutorials/htu21d-humidity-sensor-hookup-guide>
- Arduino YunMini-ren funtzionamendua (2016),
<http://www.arduino.org/products/boards/arduino-yun-mini>
- Arduino ProMini-ren funtzioanmendua (2016),
<https://www.arduino.cc/en/main/arduinoboardpromini>
- Navarro, K. (2014), nokia 5510,
<http://www.panamahitek.com/nokia-5110-arduino/>
- Casquero, O. (2016), Web sistemak,
<http://tic-investigacion.blogspot.com.es/2016/12/20161220-bloque-1-sistemas-web.html>
- Carrasco, M. (2012), XBee-n komunikazio konfigurazioa,
<http://www.internetdelas cosas.cl/2012/08/26/primeros-pasos-en-xbee-usando-arduino/>
- XBee-n tutoriala (2009),
<https://www.sparkfun.com/search/results?term=xbee&tab=tutorials>



- HCF 4050 datasheet,

<http://pdf.datasheetcatalog.com/datasheet2/0/01slcj14uf4688drxqfuui427d3y.pdf>

- LCD nokia pantaila,

<http://teslabem.com/pantalla-lcd-nokia-de-48-x-84-3310.html>

- Pantailan idazteko u8glib aginduak,

<https://github.com/olikraus/u8glib/wiki/userreference#getfontascent>

- Arduinoa lokartzeko prestatu,

<http://playground.arduino.cc/learning/arduinossleepcode>

- Arduinoa bateriaz elikatzeko,

<https://www.luisllamas.es/alimentar-arduino-baterias/>

- FTDI USB konexioak, <https://www.sparkfun.com/products/9873>

- XBee adaptadore datasheet,

<https://www.parallax.com/sites/default/files/downloads/32400-XBee-USB-Adapter-Documentation-v1.0.pdf>

- PCD 8544 lcd pantaila datasheet,

<https://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf>