



Universitat Oberta de Catalunya

Enginyeria Tècnica de Telecomunicacions

Especialitat Telemàtica

Treball Final de Carrera

Àrea SIG

Monitorització Trailwalker

Alumne: Montse Baldomà Cases

Director: Francesc Puigvert Pell
Professor responsable: Antoni Pérez Navarro

Curs 2012-2013 (setembre)

Llicència (Creative Commons)

Aquest treball està subjecte -excepte que s'indiqui el contrari- a una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de Creative Commons. Podeu copiar-lo, distribuir-lo i transmetre'l públicament sempre que en citeu l'autor i l'obra, no se'n faci un ús comercial i no se'n faci còpia derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>.

Agraïments

Primerament vull donar les gràcies a en Francesc Puivert Pell. He gaudit del seu recolzament, simpatia i bon humor i, sobretot, d'una predisposició màxima a ajudar-me a superar qualsevol entrebanc. Podria citar una infinitat de detalls: els vídeos que va preparar per tots els alumnes de l'aula per tal de facilitar-nos l'entrada en l'extens món d'ArcGIS, els nombrosos *hangouts* a què ens ha convidat a participar, les revisions detallades i exhaustives de tots els lliuraments provisionals de les PACS, les posteriors revisions de les PACS i els corresponents comentaris encoratjadors, les innombrables respostes a la immensa quantitat de preguntes que li he anat plantejant, les constants orientacions en relació als continguts i el format del treball, etc.

També vull agrair al responsable de l'aula, en Toni Pérez Navarro, l'esforç que ha fet perquè el material arribés amb suficient temps per poder realitzar el treball, malgrat tots els endarreriments i entrebancs imprevisibles que ha patit.

Finalment, he de donar les gràcies a tota la meva família, el meu pare Joan, la meva mare Elena i el meu fill Jordi, perquè tots ells s'han sacrificat d'una manera o altra perquè pogués realitzar aquest treball.

Índex

Agraïments	ii
Índex	iii
Índex de figures	iii
Índex de taules	iii
1 Introducció	1
1.1 Monitorització Trailwalker	1
1.2 GPS	1
1.3 Aplicacions del GPS	2
1.3.1 Seguiment d'objectes o persones	2
1.3.2 El problema: Monitorització Trailwalker	3
2 El dispositiu mòbil	4
2.1 Justificació de l'elecció d'Arduino	4
2.2 La placa Arduino UNO rev3	6
2.2.1 Memòria	6
2.2.2 Alimentació	7
2.2.3 Entrades i sortides	7
2.2.4 Comunicació	8
2.2.5 La programació d'Arduino	8
2.3 La placa 3G/GPRS amb GPS intern	9
2.3.1 El mòdul SIM5218	10
2.3.2 Funcionalitats de la placa 3G/GPRS amb GPS intern	10
2.3.3 GPS intern	11
2.4 Connexió de la placa 3G/GPRS i la placa Arduino Uno	12
2.4.1 Pins i funcions utilitzades	12
2.5 Aplicació	15
2.5.1 Funcions d'Arduino utilitzades	15
2.5.2 Comandes AT	15
2.5.3 Definició d'IMSI i justificació de l'elecció	19
2.6 Avaluació del consum	19
2.6.1 Estimació de la intensitat	19
2.6.2 Mesures per reduir la intensitat	19

2.6.3	Estimació de la intensitat mitjana i del tipus de bateria	22
2.6.4	Estimació de la intensitat mitjana mitjançant una pila HR22	24
3	Serveis Web	27
3.1	SOAP	27
3.2	REST	28
3.3	Avantatges de REST	29
4	ArcGIS	31
4.1	Introducció	31
4.1.1	Crear mapes intel·ligents	31
4.1.2	Compilar informació geogràfica	31
4.1.3	Crear i gestionar bases de dades geogràfiques	32
4.1.4	Resoldre problemes amb anàlisi espacial	32
4.1.5	Crear aplicacions basades en mapes	32
4.1.6	Comunicar i compartir informació	32
4.2	Mòduls d'ArcGIS	33
4.2.1	ArcGIS Desktop	33
4.2.2	ArcGIS Server	33
4.2.3	ArcGIS Online	33
4.3	Servei REST	34
4.3.1	El servei de directoris i l'API REST	34
4.4	Integració del dispositiu mòbil amb el servei REST	35
4.4.1	Formularis	35
4.4.2	JSON	36
4.4.3	Protocol HTTP: POST d'un objecte JSON	37
4.5	L'API d'ArcGIS per JavaScript	38
4.5.1	Dojo	38
4.5.2	Dojo dijts	40
4.5.3	Dojo layout dijts	40
4.5.4	Dades amb coneixement del temps	41
4.6	El visor Web	42
5	Línies futures de desenvolupament	44
5.1	Dispositiu mòbil	44
5.2	Servei Web	45
5.3	Visor Web	46
6	Conclusions	47
6.1	Objectius assolits	47
6.2	Depuració	48
6.2.1	Disseny del dispositiu mòbil	48
6.2.2	El mòdul SIM5218	49
6.2.3	Mida de la SRAM	49
6.3	Valoració de la planificació	50

A	Appèndix	51
A.1	Mesures a la terra	51
A.1.1	Radi de la terra	51
A.1.2	Distància entre dos punts de la terra	52
	Bibliografia	53

Índex de figures

2.1	Microprocessador Arduino Uno rev3	6
2.2	Diagrama de la placa 3G/GPRS amb GPS intern	9
2.3	Placa Arduino Uno connectada amb placa 3G/GPRS.	12
2.4	Placa 3G/GPRS (perspectiva 1).	13
2.5	Placa 3G/GPRS (perspectiva 2).	13
2.6	Corrent subministrada en funció de la freqüència. [8]	20
2.7	Fragment de l'esquemàtic de la placa 3G/GPRS (Mòdul SIM5218)	21
4.1	Servei REST: operació <i>Add Features</i> (realitzada mitjançant el mètode POST).	36
4.2	Captura realitzada amb l'aplicació Wireshark d'una petició HTTP POST	37
4.3	Aplicació <i>Web</i> de seguiment dels equips	42
A.1	Triangle esfèric.	52

Índex de taules

2.1	Opció 1 pel cost del dispositiu mòbil de seguiment.	5
2.2	Opció 2 pel cost del dispositiu mòbil de seguiment.	5
2.3	Intensitat dependent dels mòduls actius (cas 9 V).	19
2.4	Avaluació de la intensitat mitjana.	23
2.5	Piles AA necessàries en el cas d'adormir el microcontrolador i el mòdul.	23

Capítol 1

Introducció

1.1 Monitorització Trailwalker

El Trailwalker és una de les eines d'Intermon Oxfam de recaptació de fons. Consisteix en recórrer 100 km caminant o corrent en equips de 4 persones juntes (no són relleus) en un màxim de 32 hores. Per participar, els equips han de recaptar un mínim de 1500 euros. El 20 d'abril del 2013 es celebra la tercera edició i s'espera una participació de 400 equips. Tot el que es recapti va al fons comú d'Intermon Oxfam i es distribueix entre projectes i països segons les seves necessitats. Es necessita conèixer la ubicació en temps real de cada equip per dur a terme una monitorització durant la marxa. Els requeriments són:

- La solució ha de ser econòmica, pràctica i sostenible, de manera que per a Intermon Oxfam la implementació sigui senzilla i els costos de manteniment baixos.
- La informació derivada de la solució ha d'aportar la localització dels equips.
- La informació derivada de la solució ha d'aportar la classificació dels equips (*ranking*).
- La informació ha de estar disponible, no només pels organitzadors de l'esdeveniment, sinó que també pels marxadors, equips de suport i públic en general. Per tant, ha de ser visible des de diversos dispositius mòbils i, a ser possible, a través de la Web de Trailwalker.
- La solució ha de comptar amb propostes d'emissió de senyal. Els participants no compten amb un xip en l'actualitat, i en el cas que els corredors portessin un dispositiu emissor, aquest no ha de suposar molèsties, ni a nivell de pes, ni de càrrega de bateria, etc.

1.2 GPS

El GPS (*Global Positioning System*) o sistema de posicionament global és un sistema de posicionament global de navegació per satèl·lit (GNSS) que permet determinar en tot el món la posició de qualsevol objecte amb una precisió de pocs metres o, fins i tot, de centímetres si s'utilitza el GPS diferencial. El sistema fou desenvolupat pel Departament de Defensa dels Estats Units i està format per una xarxa de 24 satèl·lits en òrbita aproximadament a 26.560 km respecte el centre de la terra. Les seves trajectòries estan sincronitzades i cobreixen tota la superfície de la terra, de manera que des de qualsevol punt del món i en qualsevol moment seran visibles un mínim de 4 satèl·lits.

El sistema de coordenades que utilitza és el Sistema Geodèsic Mundial 1984 (WGS84). Per posicionar un objecte es necessita un mínim de tres satèl·lits de la xarxa, de cadascun dels quals es reben senyals amb un codi d'identificació del satèl·lit i l'hora del seu rellotge atòmic. A partir d'aquestes senyals, l'aparell receptor sincronitza el rellotge del GPS i calcula el temps que triguen a arribar aquests senyals. D'aquesta manera es pot esbrinar a quina distància es troba l'aparell de cada satèl·lit. Com que, a més a més, cada satèl·lit emet la posició en què es troba, es pot determinar aleshores fàcilment la posició de l'aparell a partir de les distàncies obtingudes a cadascun dels tres satèl·lits. A més, amb el senyal rebut d'un quart satèl·lit es pot determinar amb molta precisió l'hora.

La precisió en el posicionament es veu afectada per:

- Endarreriment del senyal en la ionosfera i la troposfera.
- Senyal multirruta, produïda pel rebot del senyal en muntanyes i edificis propers.
- Imprecisió en les dades de l'òrbita del satèl·lit.
- Nombre i geometria dels satèl·lits visibles.
- Errors locals en el rellotge de l'aparell receptor GPS.

El satèl·lit emet dos senyals diferents de radiofreqüències: el civil (L1) amb una freqüència portadora de 1575,42 MHz i el militar (L2) amb una freqüència portadora de 1227,60 MHz (L2). Ambdós senyals estan codificats amb claus diferents. Enviant dos senyals amb freqüències diferents, s'aconsegueix quantificar l'endarreriment del senyal en la ionosfera i la troposfera. L'exactitud del sistema GPS per ús civil té una mitjana de 15 m, i un màxim de 2.5 m. En canvi, per ús militar, l'exactitud és com a màxim de 30 cm. Es preveu en el futur la incorporació d'un nou senyal L2 per l'ús civil.

1.3 Aplicacions del GPS

Les aplicacions civils que fan ús del GPS són múltiples: navegació terrestre, marítima i aèria, topografia, construcció, localització agrícola, ramadera i de fauna, salvament i rescat, esport i oci, localització de malalts, discapacitats i menors, seguiment i recuperació de vehicles,... Aquest treball es centrarà en una d'aquestes aplicacions: el seguiment.

1.3.1 Seguiment d'objectes o persones

Quan es pensa en la paraula seguiment, un dels primers exemples que ve al cap es troba en l'àmbit de l'espionatge, on el seguiment és sense consentiment. Evidentment que hi ha altres casos on el seguiment serà amb consentiment o necessari: localització de malalts, discapacitats i menors, seguiment i recuperació de vehicles, seguiment de participants en curses de llarg recorregut,...

Per poder dur a terme una aplicació de seguiment, es requereix la ús del GPS conjuntament amb un sistema per enviar dades com, per exemple, una connexió GPRS. Les dades de posició s'envien a una adreça IP pública d'un servidor. Es pot utilitzar el protocol TCP. Les dades rebudes posteriorment s'han de visionar mitjançant alguna aplicació amb els mapes de fons necessaris que permetin localitzar, no tan sols posicionar, la persona o l'objecte fàcilment.

Per tant, a nivell de maquinari, es necessita un dispositiu mòbil amb funcionalitats GPS i GPRS i un servidor connectat a Internet. A nivell de *software* es necessita l'aplicació del dispositiu mòbil, l'aplicació del servidor i l'aplicació per visionar les dades. No s'ha d'oblidar tota la cartografia necessària de fons perquè, en visionar les dades rebudes, es pugui localitzar directament l'objecte o la persona.

Existeixen moltes opcions possibles pel dispositiu mòbil:

- Actualment els mòbils, ja no serveixen només per realitzar trucades i enviar missatges SMS, sinó que incorporen un ampli ventall de funcionalitats que inclouen GPS i GPRS. És a dir, un d'aquests mòbils pot ser configurat directament o mitjançant una aplicació per tal que realitzi la funció de dispositiu mòbil de seguiment. És important destacar que les bateries d'aquests dispositius són de curta durada. Pot reduir-se a 4 o 5 hores si l'ús que se'n fa és molt intensiu.
- Altres dispositius han estat específicament dissenyats per ser dispositius mòbils de seguiment i és aquesta l'única funcionalitat que tenen. S'utilitzen en espionatge, seguiment de vehicles, localització d'infants,.... Es pot elegir el model en funció de l'ús. Per exemple, si l'objectiu és el seguiment de vehicles s'elegirà un model amb uns forts imans per adherir-se a un automòbil sense necessitat de cargols. Per altra banda, existeixen models amb bateries que permetrien fer un seguiment durant 6 mesos. La durada de les bateries és també un paràmetre a valorar en l'elecció del model de dispositiu.

1.3.2 El problema: Monitorització Trailwalker

Es tracta d'un problema de seguiment complet:

- Maquinari
 - Elecció del dispositiu mòbil de seguiment que sigui adequat a les condicions i durada de la cursa i que sigui econòmic.
 - Disponibilitat de dos servidors amb IP pública: un servidor per recollir dades i l'altre per visionar-les.
- Programari
 - Programació de l'aplicació pel dispositiu mòbil elegit.
 - Elecció de la font per la cartografia de fons.
 - Programació de l'aplicació que permetrà fer un seguiment dels equips. Es tractarà d'una aplicació en llenguatge HTML 4 JavaScript i el seguiment serà possible a través d'una connexió a Internet.

Capítol 2

El dispositiu mòbil

2.1 Justificació de l'elecció d'Arduino

En l'apartat 1.3.1 he fet referència a dos tipus de dispositius mòbils de seguiment: els genèrics i els específics. Amb genèrics em refereixo als mòbils més actuals que ja formen part de la vida quotidiana de molts i que tenen un munt de funcionalitats. Amb específics em refereixo a dispositius dissenyats específicament per permetre'n el seguiment. Analitzant amb més detall les diferents opcions es veu que:

- L'opció més econòmica i de configuració més senzilla és la primera: els genèrics. És l'opció més econòmica per Intermon Oxfam, ja que els dispositius són propietat dels participants. A més, és força probable que els participants ja disposin d'un dispositiu d'aquest tipus. Per tant, tampoc suposarà una inversió pel participant. El problema principal d'aquests tipus de dispositius és la durada de la bateria. Aquesta, podria arribar a ser de 4 a 5 hores si se'n fa una ús molt intensiu. La durada màxima de la cursa és de 32 hores i els equips de 4 persones. En aquest cas el problema s'ha de resoldre per un altre camí. Es tracta d'un problema logístic:
 - Quants mòbils té l'equip?
 - Quantes bateries de recanvi té l'equip?
 - Estudi de la configuració per tal que el consum sigui mínim. Per posar algun exemple, en estat de repòs l'iPhone 5 pot durar fins a 225 hores, el Samsung Galaxy S III fins a 590 hores.
 - Estudi de si és necessari o no apagar el mòbil en un cert interval de temps.
 - Estudi de la periodicitat en què el GPS recull la posició i s'envia a l'URL de destí.

Per tant, en aquest cas els participants hauran d'estar informats de com han d'utilitzar els mòbil durant la cursa per tal que el seguiment és pugui realitzar amb èxit. Aquest és un factor de risc perquè les recomanacions es poden seguir o no.

- Amb l'elecció d'un dispositiu específic s'aconsegueix resoldre el problema de la durada de la bateria. Molts d'aquests dispositius estan dissenyats per fer un seguiment llarg, mesurat en mesos, no hores. Són dispositius relativament petits, ja que en molts casos s'utilitzen en tasques d'espionatge. El seu desavantatge principal és el preu. El model GPS HI-604X,

de l'empresa HICOM, té una autonomia de fins 5 anys en mode repòs. Les seves dimensions són de 5.2 cm × 9.1 cm × 3.9 cm. El seu preu no és molt elevat. Un dels preus més baixos per aquest model és de 235 € sense IVA a <http://www.paradetectives.es/gps/433-gps-5-anos-autonomia-hi-604x.html>.

- Existeix una opció més didàctica i més econòmica que l'opció anterior. En moltes aplicacions en lloc de construir circuits a partir de components bàsics, es poden utilitzar microcontroladors. És a dir, el *software* substitueix el disseny electrònic. Per tal de construir el dispositiu de seguiment es necessitarà bàsicament un microcontrolador, un mòdul GPS i un mòdul GPRS. La placa Arduino incorpora un microcontrolador i té petites dimensions (68.6 mm × 53.3 mm). Es pot utilitzar per construir dispositius interessants com aquests. És molt econòmica. Es pot comprar per 20 € a <http://www.cooking-hacks.com/index.php/arduino-uno.html>. En la taula (2.1) es pot veure el preu total del dispositiu i en la taula (2.2) el preu total quan els mòduls GPS i GPRS estan integrats en una sola placa. La placa amb el GPS i GPRS integrats és més cara, ja que té moltes funcionalitats:
 - El GPS intern permet el posicionament a l'exterior i a l'interior ja que combina les trames NMEA (*National Marine Electronics Association*) estàndard amb la triangulació d'identificador de cel·la de mòbil. Pot utilitzar el mode mòbil assistit (A-GPS) i el mode basat en el mòbil (S-GPS).
 - És compatible amb xarxes 3G HSPA (*High Speed Packet Access*) i WCDMA (*Wideband Code Division Multiple Access*). Pot funcionar també com a mòdem 3G estàndard en Linux//Windows/MacOS, funcionant a plena velocitat.
 - Es poden connectar a la placa: videocàmera, micròfon, altaveu, auriculars. La placa també té una ranura per targeta microSD.

Placa Arduino Uno rev3	20 €
Modul GPS per Arduino	42 €
Antena GPS interna	8 €
Mòdul Quadband GPRS/GSM per Arduino (SIM900)	63 €
Antena GPRS interna	6 €
Total	139 €

Taula 2.1: Opció 1 pel cost del dispositiu mòbil de seguiment.

Una altra opció seria

Placa Arduino Uno rev3	20 €
Placa 3G/GPRS amb GPS intern	149 €
Antena GPS interna	8 €
Antena GPRS-GSM-UMTS externa	6 €
Total	183 €

Taula 2.2: Opció 2 pel cost del dispositiu mòbil de seguiment.

En aquest treball s'optarà per l'opció de la placa Arduino UNO i la placa 3G/GPRS amb GPS intern. Tot i que no és l'opció més econòmica, s'espera que la durada de la bateria sigui suficient

per tota la cursa. També és una opció didàctica i no s'ha d'oblidar que aquest és un objectiu fonamental en qualsevol treball de final de carrera.

2.2 La placa Arduino UNO rev3

En la figura (2.1) es pot veure una placa Arduino UNO rev3. Aquesta placa conté un microcontrolador ATmega328, 14 pins digitals d'entrada/sortida, 6 dels quals poden ser utilitzats com sortides PWM, 6 pins d'entrada analògics, un cristall de quars oscil·lador a 16 MHz, una connexió USB, un connector d'alimentació, una capçalera ICSP i un botó *reset*. La longitud i amplada de la placa Arduino són 68.58 i 53.34 mm respectivament. Els connectors USB i d'alimentació sobresurten respecte aquest rectangle. La placa es pot carregar amb 4 cargols a un panell o a una caixa.

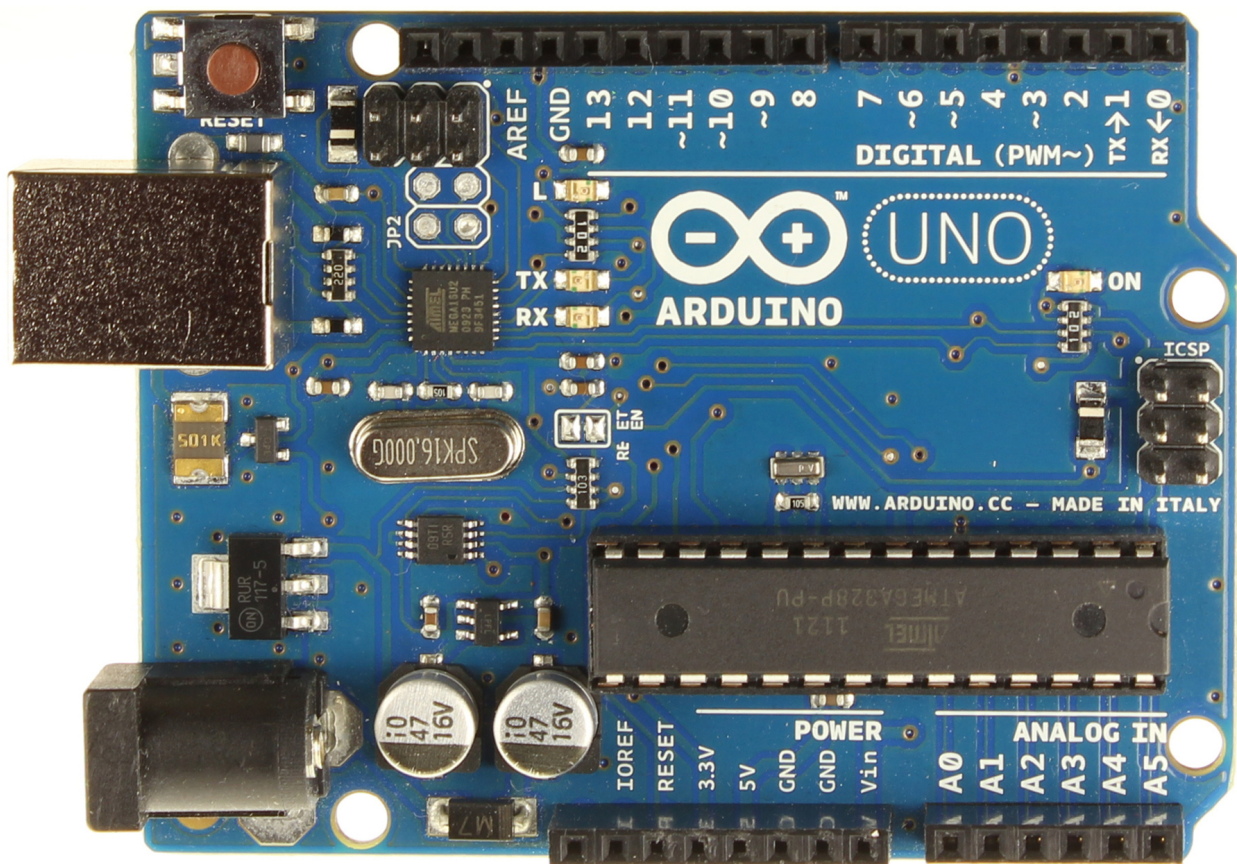


Figura 2.1: Microprocessador Arduino Uno rev3

2.2.1 Memòria

La placa Arduino disposa de tres tipus de memòria:

- Una memòria Flash de 32 KB, dels quals el *bootloader* utilitza 0.5 KB. En aquesta memòria és on es carrega l'executable. Suporta 10000 escriptures o càrregues d'executables.
- Una memòria SRAM (*Static random-access memory*) de 2 KB, que és la que fa servir el programa o *sketch* durant l'execució. Aquesta memòria és volàtil. El que hi ha emmagatzemat és perd quan es desconnecta la font d'alimentació. Es pot llegir i escriure en aquesta memòria de

manera molt ràpida. Les variables normals de l'*sketch* s'emmagatzemen en aquesta memòria mentre l'*sketch* s'està executant.

- Una memòria EEPROM (*Electrically Erasable Programmable Read-Only Memory*) de 1 KB. Aquesta és una tecnologia antiga que implementa memòria reescribible no volàtil. S'utilitza per emmagatzemar configuracions i paràmetres constants per diferents execucions. Suporta un mínim de 100000 escriptures. Es disposa de la llibreria EEPROM d'Arduino per llegir i escriure en aquesta memòria.

2.2.2 Alimentació

La font d'alimentació pot ser mitjançant una adaptador AC/DC, una bateria o mitjançant una connexió USB amb un ordinador. La selecció de la font es realitza de manera automàtica. L'endoll que es connecta al connector d'alimentació és un de tipus 21 mm centre-positiu. Els pols de la bateria poden inserir-se en els pins GND i Vin del connector d'alimentació. Els ports USB de l'ordinador romanen protegits davant de possibles pics de corrent. Si s'apliquen més de 500 mA al port USB, un fusible automàticament interromprà la comunicació fins que el curtcircuit o la sobrecàrrega hagi estat eliminada.

La placa pot operar amb una tensió d'entrada de 6 a 20 V, però es recomana que sigui de 7 a 12 V. Si es subministra menys de 7 V, aleshores el pin que subministra 5 V, pot subministrar-ne menys i la placa pot esdevenir inestable. Si el voltatge és superior a 12 V, el regulador de voltatge es pot sobreescalfar i danyar la placa.

Els pins d'alimentació són els següents:

- Vin. És l'entrada de voltatge per la placa Arduino. Es pot subministrar voltatge per aquest pin, o bé, si s'està subministrant voltatge pel connector d'alimentació, accedir-hi per aquest pin.
- 5 V. Aquest pin dona 5 V regulats pel regulador de la placa. Aquest pin es fa servir de sortida. La placa es pot alimentar amb: corrent DC pel connector d'alimentació (7-12 V), el connector USB (5 V) o bé el pin Vin de la placa (7-12 V). Subministrar tensió pels pins 5 V o 3.3 V eviten el pas pel regulador de tensió i es pot danyar la placa. No s'aconsella.
- 3.3 V. El regulador de la placa pot subministrar 3.3 V, amb una intensitat màxima de 50 mA.
- GND. Pin de terra.

2.2.3 Entrades i sortides

Té 14 pins digitals d'entrada/sortida, 6 dels quals poden ser utilitzats com sortides PWM. Cadascun dels 14 pins digitals poden ser utilitzats com entrada o sortida utilitzant les funcions `pinMode()`, `digitalWrite()` i `digitalRead()`. Aquests pins operen a 5 V. Cadascun d'ells pot generar o rebre un màxim de 40 mA de corrent DC. Algun d'aquests pins estan especialitzats:

- Serial: 0 (RX) i 1 (TX). RX s'utilitza per rebre dades en sèrie TTL (*Transistor-Transistor Logic*) i TX per transmetre-les.

- Interrupcions externes: 2 i 3. Aquests pins estan configurats per disparar, donades unes certes condicions, una interrupció. La funció `attachInterrupt()` en permet la programació.
- PWM: 3, 5, 6, 9, 10 i 11. Aquest bits poden ser utilitzats com sortides PWM. PWM significa *Pulse Width Modulation* o Modulació d'Amplada de Pols. Les sortides PWM en lloc d'emetre un senyal continu, emeten una sèrie de polsos a freqüència constant. Depenent de l'amplada del pols la tensió mitjana resultant pot variar de valor. És a dir, aquests pins digitals poden emular una sortida analògica. Es pot generar una sortida PWM de 8 bits amb l'ajuda de la funció `analogWrite()`.
- SPI (*Serial Peripheral Interface Bus*): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Aquests pins suporten la comunicació SPI, utilitzant la llibreria SPI.
- LED: 13. Hi ha un LED connectat al pin digital 13. Quan el pin té el valor HIGH, el LED està encès. En canvi si el pin pren el valor LOW, el LED està apagat.

Té 6 entrades analògiques etiquetades A0, A1, A2, A3, A4, A5. Cadascuna d'elles té una resolució de 10 bits, és a dir, 1024 valors diferents. Per defecte, mesuren tensions de fins a 5 V. Es pot canviar aquest límit superior utilitzant el pin AREF i la funció `analogReference()`. Alguns d'aquests pins estan especialitzats. El pins que suporten la comunicació TWI (*Two Wire Interface*) utilitzant la llibreria `Wire` són A4 (pin SDA) i A5 (pin SCL).

Hi ha una correspondència entre els pins d'Arduino i els ports del microcontrolador ATmega328.

2.2.4 Comunicació

La placa Arduino UNO pot connectar-se amb un ordinador, amb un altra placa Arduino o amb un microcontrolador. El microcontrolador ATmega328 proporciona una comunicació en sèrie UART (*Universal Asynchronous Receiver/Transmitter*) TTL a 5 V (nivell baix: 0 V, nivell alt: 5 V), disponible en els pins digitals 0 (RX) i 1 (TX). El xip ATmega16U2 canalitza la comunicació sèrie via USB i apareix com un port COM virtual en el *software* de l'ordinador. Els *drivers* utilitzats són *drivers* per COM USB estàndard i no es necessita cap *driver* exterior. No obstant això, a Windows fa falta un arxiu `.inf`. El *software* d'Arduino inclou un monitor que permet enviar i rebre dades a la placa Arduino. Els leds RX i TX s'encendran quan les dades es transmeten pel xip ATmega16U2 i la connexió USB amb l'ordinador, però no per la comunicació sèrie a través dels pins 0 i 1. La llibreria `SoftwareSerial` permet aquesta comunicació sèrie.

El microcontrolador ATmega328 permet també la comunicació SPI i I²C (*Inter-Integrated Circuit*)¹. S'usen les llibreries `SPI` i `Wire`, respectivament.

2.2.5 La programació d'Arduino

L'Arduino UNO es pot programar amb el *software* d'Arduino. En la memòria Flash hi ha preinstal·lat un carregador d'arranc o *bootloader* que permet carregar codi nou sense l'ús d'un programador de *hardware* exterior. Alternativament es pot programar l'Arduino sense utilitzar el *bootloader*, sinó mitjançant la capçalera ICSP (*In-Circuit Serial Programming*).

¹I²C és un bus de comunicacions sèrie molt utilitzat per comunicar microcontroladors i els seus perifèrics en sistemes integrats (*Embedded Systems*). TWI és un bus idèntic a I²C. El nom de TWI va ser introduït per Atmel i altres companyies per evitar conflictes de patents.

En lloc de prémer el botó *reset* abans de carregar un nou *sketch*, l'Arduino està dissenyat per tal que aquest *reset* el realitzi el *software* executant-se en l'ordinador. Només cal clicar el botó *Upload* en l'entorn de programació d'Arduino i es realitza un *reset* automàtic.

Un programa d'Arduino s'anomena *Sketch*, que vol dir esbós. L'Arduino es programa en un derivat del llenguatge de programació *Wiring*. La seva estructura i sintaxis és igual a la de C. Es poden utilitzar totes les llibreries de C i algunes de C++. L'estructura bàsica d'un *Sketch* és molt similar a la de qualsevol programa en C amb algunes diferències ja que aquesta plataforma ha d'interactuar amb el medi físic que l'envolta. La primera part d'un *Sketch* és la declaració de variables, on es defineixen les variables globals declarant el seu tipus i, si es vol, el seu valor inicial. Aquestes variables seran accessibles des de qualsevol part l'*Sketch*, a diferència de les variables locals, que es declaren dins d'una funció o una estructura de control, com per exemple un bucle.

La segona part d'un *Sketch* és una part que s'executa només una vegada després d'arrancar la placa Arduino. Es tracta de la funció `setup()`. Sovint s'utilitza per definir els *pins* que s'utilitzen i declarar-los com a entrades o sortides. Tot i així, al llarg de l'*Sketch* es pot canviar la definició dels pins. La següent part important d'un *Sketch* és la funció `loop()`. És la part principal del programa i conté el codi que el microcontrolador executarà recursivament, mentre la placa Arduino disposi d'energia.

2.3 La placa 3G/GPRS amb GPS intern

En la figura (2.2) es pot veure el diagrama de la placa 3G/GPRS amb GPS intern. La placa

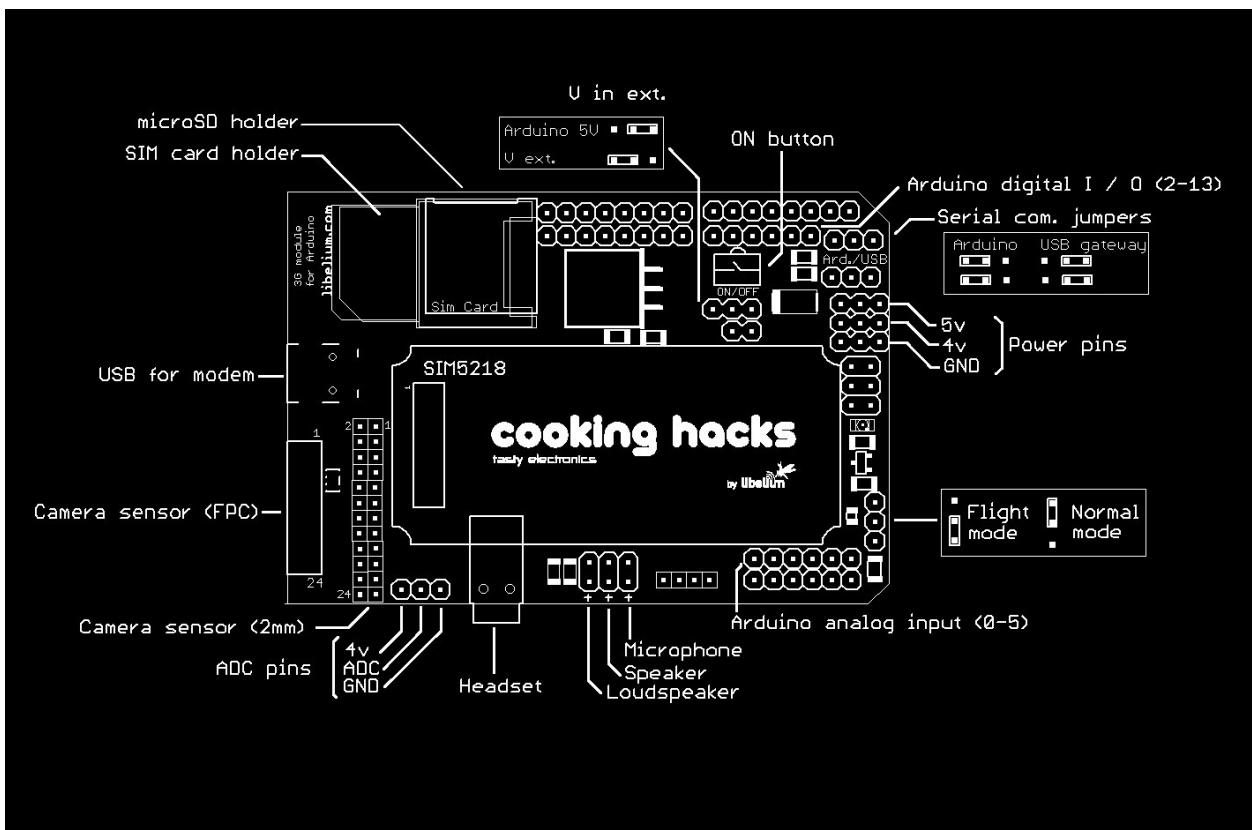


Figura 2.2: Diagrama de la placa 3G/GPRS amb GPS intern

3G/GPRS s'alimenta d'una única font d'alimentació de 3.4 V a 4.2 V i es basa en un mòdul de la

sèrie SIM5218 de SIMCOM. El rang de voltatge d'entrada d'aquest mòdul és també de 3.4 a 4.2 V.

2.3.1 El mòdul SIM5218

El mòdul SIM5218 és una solució UMTS/HSPA tribanda (900/1900/2100 MHz) i una solució GSM/GPRS/EDGE quadribanda (850/900/1800/1900 MHz). Suporta serveis de fins 7.2Mbps de velocitat de baixada i 5.76Mbps de pujada. Té un gran ventall de capacitats i interfícies: UART, USB2.0 d'alta velocitat, una targeta SIM incrustada, targeta SD, càmera, GPS, etc. Té moltes possibilitats d'aplicació, tals com TCP/UDP/FTP/FTPS/HTTP/HTTPS/SMTP/POP3 i MMS. Es controla mitjançant comandes AT. Les velocitats màximes assolides són:

- HSPA (*High-Speed Packet Access*):
 - 7.2 Mbps de baixada (HSDPA)
 - 5.76 Mbps de pujada (HSUPA)
 - 3.6 Mbps de baixada i 2.0 Mbps de pujada (HSDPA + HSUPA)
- UMTS (*Universal Mobile Telecommunications System*)
 - 384Kbps de baixada
 - 384Kbps de pujada
- EDGE Class
 - 236.8 Kbps de baixada
 - 118 Kbps de pujada
- GPRS (*General Packet Radio Service*)
 - 85.6 Kbps de baixada
 - 42.8 Kbps de pujada

El mòdul SIM5218 té 3 connectors U.FL (connectors RF en miniatura), destinats a:

- Antena GPS
- Antena principal per la connexió mòbil.
- Antena secundària per permetre la diversitat. Només s'utilitza si s'experimenten problemes de cobertura o rendiment.

2.3.2 Funcionalitats de la placa 3G/GPRS amb GPS intern

El GPS intern permet la localització del dispositiu a l'exterior i a l'interior d'edificis, gràcies a què combina les trames NMEA estàndard amb la triangularització efectuada a partir de tres estacions bases. El GPS té tres modes de funcionament: mode independent, mode mòbil assistit (A-GPS) i el mode basat en el mòbil (S-GPS). La localització que dona el GPS mitjançant sentències NMEA pot ser completada per la informació de la cel·la subministrada pel mòdul 3G i per servidors de geoposició d'Internet externs. D'aquesta manera la localització es més acurada.

La placa està especialment orientada a treballar amb servidors d'Internet. Es pot navegar per Internet mitjançant els protocols HTTP i HTTPS i així baixar contingut d'un servidor Web o pujar-n'hi. Es poden utilitzar els protocols FTP, FTPS en aplicacions que requireixen manejar arxius. A més a més, es poden enviar i rebre correus electrònics utilitzant un client de SMTP i POP3 implementats internament.

Es poden connectar a la placa altres accessoris com: videocàmera, micròfon, auriculars, altaveu intern i altaveu extern. També disposa d'una ranura per targetes de memòria SD. Fins i tot pot reproduir arxius d'àudio comprimit. És pot utilitzar com a mòdem 3G a plena velocitat connectant-lo directament a un ordinador (Linux, Windows, MacOS) a través de la seva ranura mini-USB. L'objectiu de la ranura per targetes de memòria SD és poder manejar un sistema de fitxers FAT16 que pugui emmagatzemar fins 32 GB d'informació. El mòdul 3G pot treballar amb arxius en la memòria SD sense necessitat d'Arduino per la seva manipulació.

Hi ha moltes accions que es poden fer en mòdul 3G sense haver de fer servir el microcontrolador Arduino. Tot i que el programa s'executi en l'Arduino les dades poden viatjar des del mòdul 3G a la targeta SD o des de la videocàmera al mòdul 3G sense passar per l'UART. Així s'assegura arribar a la màxima velocitat possible que la xarxa de dades de la companyia telefònica associada a la targeta SIM pot donar, que com a màxim és aproximadament 7.2 Mbps de baixada i 5.76 Mbps de pujada. Es poden baixar arxius directament a la targeta SD des de servidors HTTP o FTP o, a l'inrevés, pujar-los-hi. Es pot gravar àudio o vídeo i emmagatzemar-lo a la targeta SD o enviar-lo directament al servidor.

L'altra opció és utilitzar l'UART d'Arduino a plena velocitat, és a dir, aproximadament, 115 kbps. Aquesta és la mateixa velocitat que s'obté si connectes l'Arduino pel port USB. És més que suficient en el cas d'utilitzar la xarxa GPRS.

Es pot controlar el tipus de servei de xarxa i la qualitat del senyal rebut de les cel·les al voltant i es pot escollir entre GSM, GPRS, UMTS i HSPA depenent de la cobertura.

2.3.3 GPS intern

El GPS convencional té dificultats a l'hora de proporcionar posicions en condicions de senyal baixa: edificis alts, interiors d'edificis, sota els arbres,... A més, el TTFF (*Time To First Fix*) sol ser molt llarg i, depenent de les condicions, pot ser de minuts. En un mode de funcionament GPS assistit, el dispositiu utilitzarà dades obtingudes d'un servidor extern i ho combinarà amb la informació de la cel·la de telefonia mòbil per conèixer la posició i saber quins satèl·lits té a sobre. Aquesta informació la completarà amb la informació rebuda a través del GPS convencional. El TTFF es reduirà notablement i el posicionament serà més acurat.

SIM5218 suporta tres modes de funcionament. En el mode A-GPS (assistit al mòbil) és el servidor extern qui calcularà la posició del dispositiu mòbil. En el mode S-GPS (basat en el mòbil) és el propi dispositiu mòbil qui calcularà la pròpia posició a partir de tota la informació recollida. En el mode independent, només s'utilitzarà la informació subministrada pel GPS.

Les solucions assistides tenen un rendiment baix en zones rurals on la densitat de les cel·les és baixa. En canvi, les solucions no assistides tenen un rendiment baix en interiors. La solució GPS del mòdul SIM5218, proporciona un TTFF òptim, és acurat i redueix la utilització de la xarxa, depenent de les condicions donades.

En el mode independent, el GPS obté la posició, altitud,... només amb la senyal dels satèl·lits. És el més lent dels tres modes. Amb la comanda AT+CGPSINFO el mòdul SIM5218 proporciona

latitud, longitud, data, temps UTC (*Universal Time Coordinated*), altitud i velocitat. Si el que interessa és una sentència NMEA, que pot ser GSV, GGA, RMC, GSA i VTG, cal utilitzar la comanda `AT+CGPSSWITCH=1` i configurar el port a 57600 bauds.

2.4 Connexió de la placa 3G/GPRS i la placa Arduino Uno

En la figura (2.3) es veu com es connecta la placa 3G/GPRS amb la placa Arduino Uno. S'observa perfectament com els pins digitals del 0 al 13 i els pins GND i AREF de la placa Arduino queden connectats a la placa 3G/GPRS. De la figura (2.4) s'obté una perspectiva diferent i es dedueix que també es connectaran els pins analògics A0, A1, A2, A3, A4, A5 i els pins 5V, GND, GND i Vin. La capçalera ICSP també queda connectada a la placa 3G/GPRS. En la figura (2.5), en la part frontal, es poden veure els 6 pins de la placa 3G/GPRS que es connectaran a la capçalera ICSP mascle de la placa Arduino.



Figura 2.3: Placa Arduino Uno connectada amb placa 3G/GPRS.

2.4.1 Pins i funcions utilitzades

Es connecta el GND de la placa Arduino amb el GND de la placa 3G. La placa 3G/GPRS obté l'alimentació a partir del pin Vin de la placa Arduino.

Per la comunicació entre la placa Arduino i un ordinador o altres dispositius, com la placa 3G/GPRS, s'utilitza la comunicació de dades en sèrie. Totes les plaques Arduino tenen com a mínim un port sèrie, conegut com UART o USART. La placa Arduino es comunica amb la placa 3G a través dels pins digitals 0 (RX) i 1 (TX). Aquests pins estan també connectats a l'adaptador USB de la placa. Per utilitzar aquests pins per comunicar-se amb la placa 3G/GPRS, és a dir, un dispositiu serie extern TTL, es connecta el pin RX de la placa Arduino amb el pin TX de la placa 3G/GPRS i el pin TX de la placa Arduino amb el pin RX de la placa 3G/GPRS.

Les funcions utilitzades per la comunicació de les dades en sèrie són:



Figura 2.4: Placa 3G/GPRS (perspectiva 1).



Figura 2.5: Placa 3G/GPRS (perspectiva 2).

- **Serial.begin(velocitat)** Estableix la velocitat de dades en símbols per segon (bauds) per la transmissió de dades en sèrie. Per comunicar-se la placa Arduino amb l'ordinador es poden

utilitzar diferents velocitats: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200. Per comunicar-se la placa Arduino amb la placa 3G s'utilitza una velocitat de 115200 bauds.

- **Serial.available()** Retorna el nombre de bytes disponibles per ser llegits pel port sèrie. Es refereix a dades ja rebudes i disponibles en el búffer de recepció del port. Té una capacitat de 128 bytes.
- **Serial.read()** Llegeix les dades entrants pel port sèrie.
- **Serial.flush()** Buidia el búffer de dades en sèrie. Qualsevol crida a les funcions `Serial.read()` o `Serial.available()` retornarà només les dades rebudes després l'última crida a `Serial.flush()`.
- **Serial.print(valor)** Imprimeix les dades al port sèrie com text ASCII. El valor a imprimir pot ser de qualsevol tipus. Els caràcters i les cadenes s'imprimeixen tal qual. Per exemple: `Serial.print('N')` imprimeix N, `Serial.print("Hola")` imprimeix Hola. No retorna res.
- **Serial.println(valor)**: És com la comanda anterior però imprimeix al final un retorn de carro (caràcter ASCII 0x0D) i un avanç de línia (caràcter ASCII 0x0A).
- **Serial.write(valor)**, **Serial.write(string)**, **Serial.write(buffer, logitud)**: Escriu dades binaries. Aquestes dades s'envien com un byte o una sèrie de bytes

El pin digital número 2 s'utilitza com a sortida per tal d'engegar el mòdul 3G/GPRS, sense necessitat de prémer el botó d'engegar de la placa 3G/GPRS. Les funcions utilitzades en aquest cas són:

- **pinMode(pin, mode)**

Configura el pin especificat per actuar com una entrada o una sortida.

- pin: Indica el número de pin, del qual vols escollir el mode.
- mode: El mode pot ser INPUT, OUTPUT o INPUT_PULLUP.

No retorna res. Exemple: `pin (2, OUTPUT)`.

- **digitalWrite(pin, valor)**

Escriu el valor de HIGH o LOW en un pin digital. Si el pin és de sortida, el seu voltatge es posarà a 5 V en el cas de HIGH i 0 V en el cas de LOW. En el cas que el pin sigui d'entrada, el valor de HIGH activarà un resistència d'activació de 20 K Ω , i el valor de LOW la desactivarà.

- pin: Indica el número de pin.
- valor: El valor pot ser HIGH o LOW.

No retorna res. Exemple: `digitalWrite (2, HIGH)`

2.5 Aplicació

2.5.1 Funcions d'Arduino utilitzades

En l'apartat 2.4.1 he comentat totes les funcions relacionades amb la comunicació de dades sèrie a través dels pins 0 i 1, i les funcions utilitzades per engegar el mòdul, mitjançant el pin 2.

Cal destacar també la funció `delay(ms)`. Aquesta funció atura el programa la quantitat de mil·lisegons especificada en el paràmetre. No retorna res. Aquesta funció provoca que gairebé tota l'activitat s'aturi: no es llegeixen els sensors, ni es fan càlculs, ni hi ha manipulació de pins. De totes maneres les interrupcions segueixen funcionant correctament, la comunicació de dades en sèrie dels pins RX segueix funcionant, els valors de PWM i els estats del pins es mantenen.

Per altra banda, la llibreria `math.h` inclou moltes funcions matemàtiques i trigonomètriques. Les funcions que s'utilitzen en l'*sketch* són: `sqrt()`, `floor()`, `sin()`, `cos()`, `acos()`. En aquests casos el paràmetre d'entrada és *double* i el valor de retorn també. La funció `sqrt()` fa referència a l'arrel quadrada, `sin()` al sinus, `cos()` al cosinus, `acos()` al arccosinus. La funció `floor()` retorna expressat com a nombre en punt flotant el major enter inferior al paràmetre d'entrada.

2.5.2 Comandes AT

- **AT+CPIN=<pin>**

Aquesta comanda envia al ME (*Mobil Equipment*) el número PIN. La resposta que es rep, en cas que el nombre sigui correcte, és OK.

- **AT+CBC**

Aquesta comanda s'utilitza per esbrinar el voltatge de la tensió subministrada. En cas de no haver error la resposta és +CBC: <bc>,<bcl>,<vol>V.

- <bc>: 0 si està alimentat per bateria.
- <bcl>: Nivell de càrrega de la bateria. Pren valors de 0 a 100.
- <vol> Tensió actual en volts.

Per exemple una resposta podria ser:

```
+CBC: 0,75,3.810V
OK
```

- **AT+CPOF**

Aquesta comanda apaga la placa 3G/GPRS. És una comanda molt important.

- A vegades la placa Arduino es reinicialitza de manera automàtica i el programari comença des de la funció `setup`, tal com s'ha explicat en 2.2.5. Si no s'apaga primerament el mòdul 3G/GPRS, aquesta reinicialització no s'efectuarà correctament i el programari es quedarà bloquejat en un bucle infinit, a l'espera de respostes que no arribaran.
- També és molt útil si es vol programar una reinicialització del mòdul perquè s'han produït errors inesperats.
- Es pot utilitzar l'opció d'apagar el mòdul 3G/GPRS per estalviar energia i allargar la durada de la bateria.

- **AT+CHTTPACT**

Inicia una operació HTTP, com podria ser GET o POST. El caràcter 0x1A indica final de les dades contingudes en la petició o en la resposta. En el cas que el caràcter 0x1A formi part de les dades, s'haurà de transmetre 0x03 0x1A. En el cas que el caràcter 0x03 formi part de les dades, s'haurà de transmetre 0x03 0x03. Aquesta comanda requereix haver introduït prèviament el número PIN.

Aquí es mostra un exemple amb peticions de l'usuari i les respostes rebudes. En aquest exemple es pot veure un diàleg que consisteix amb una petició i la seva resposta:

- **Petició**

```
AT+CHTTPACT="<address>",<port>
```

```
on
```

```
* <address>: L'adreça IP o el nom del domini del servidor.
```

```
* <port>: El port del servidor.
```

- **Resposta:**

```
+CHTTPACT: REQUEST
```

- **Petició:**

```
POST /arcgis/rest/services/TrailWalker/FeatureServer/0/addFeatures HTTP/1.1
```

```
Host: 213.73.41.144:80
```

```
Content-Length: 381
```

```
Origin: http://213.73.41.144:80
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Accept: text/html
```

```
Referer: http://213.73.41.144:80/arcgis/rest/services/TrailWalker/FeatureServer/0/addFeatures
```

```
Accept-Charset: utf-8
```

```
features=%5B%7B%22geometry%22%3A%7B%22x%22%3A%001.8321169853%2C%22y%22%3A%41.7394485473%2C%22spatialReference%22%3A%7B%22wkid%22%3A%104910%7D%7D%2C%22attributes%22%3A%7B%22CodiEquip%22%3A%214190070009393%2C%22Bateria%22%3A%065%2C%22Entrada%22%3A%222012-12-17+05%3A%42%3A%35%22%2C%22Tipus%22%3A%22Arduino%22%2C%22Equip%22%3A%22ARDUINO_000%7D%7D%5D&gdbVersion=&rollbackOnFailure=true&f=html<Ctrl+Z>
```

- **Resposta:**

```
OK
```

```
+CHTTPACT: DATA,1420
```

```
http/1.1 200 ok
```

```
cache-control: private, must-revalidate, max-age=0
```

```
content-type: text/html; charset=utf-8
```

```
etag: 2cbd8ecb
```

```
server: microsoft-iis/7.5
```

```
x-xss-protection: 1; mode=block
```

```
x-frame-options: sameorigin
```

```
access-control-allow-origin: http://213.73.41.144:80
```

```
server:
```

```
x-aspnet-version: 2.0.50727
```

```
x-powered-by: asp.net
```

```

date: thu, 13 dec 2012 19:29:56 gmt
content-length: 3034

<html>

<head>
<title>Add Features: Posicio (ID: 0)</title>
<link href="/arcgis/rest/static/main.css" rel="stylesheet" type="text/css"/>
</head>

<body>

.....

  <table class="formTable">
    <tr>
      <td><pre>{"addResults": [{
"objectId": 4006,
"success": true
}]}</pre></td>
    </tr>
  </table>

</div>

<br/><br/>

</body>

</html>
+CHTTPACT: 0

```

Les respostes d'error finalitzen amb:

– +CME ERROR

quan l'error depèn del dispositiu. Per exemple es podria donar aquest error si prèviament no s'ha introduït el número PIN.

– +CHTTPACT: <err>
ERROR

Els codis d'error per a <err> són:

- * 220 Error desconegut per a HTTP
- * 221 La tasca HTTP està ocupada
- * 222 Error en la resolució de l'adreça del servidor
- * 223 El temps per HTTP ha expirat
- * 224 Error en la transferència de dades
- * 225 Error de memòria
- * 226 Paràmetre invàlid

* 227 Error de xarxa

- **AT+CIMI**

L'equip terminal (TE) demana al adaptador de terminal (TA) l'IMSI (*International Mobile Subscriber Identity*) que identifica la targeta SIM del terminal mòbil i té com a molt 15 dígits. És necessari haver introduït prèviament el número PIN. La resposta és

```
<IMSI>
OK
```

Un exemple de resposta pot ser:

```
214190070009393
OK
```

- **AT+CGPSINFO**

Serveix per aconseguir informació de la posició mitjançant el GPS. La resposta que es rep és

```
+CGPSINFO: [<lat>], [<N/S>], [<log>], [<E/W>], [<date>], [<UTC time>], [<alt>],
[<speed>], [<course>]
OK
```

Per exemple:

```
+CGPSINFO: 3113.343286,N,12121.234064,E,250311,072809.3,44.1,0.0,0
OK
```

- <lat> Latitud de la posició actual. El format de sortida és ddmm.mmmmmm
- <N/S> Indicador Nord/Sud
- <log> Longitud de la posició actual. El format de sortida és dddmm.mmmmmm
- <E/W> Indicador Est/Oest
- <date> Data. ddmmyy
- <UTC time> Temps UTC. El format de sortida és hhmmss.s
- <alt> Altitud MSL. La unitat és en metres
- <speed> Velocitat sobre el terra. La unitat és nusos.
- <course> Direcció (graus).

- **AT+CSQ**

Retorna la mesura en dBm de la quantitat de senyal rebuda i la taxa de bits erronis del canal. La resposta té el següent format:

```
+CSQ: <rssi>,<ber>
OK
```

- <rssi> pren valors: 0 (≤ -113 dBm), 1 (-111 dBm), 2 (-109 dBm),..., 30 (-53 dBm), 31 (≥ -51 dBm), 99 (no es sap o no es detecta).
- <ber> pren valors: 0 (<0.01%), 1 (0,01%), 2 (0.1% - 0.5%), 3 (0.5% - 1.0%), 4 (1.0% - 2.0%), 5 (2.0% - 4.0%), 6 (4.0% - 8.0%), 7 ($\geq 8.0\%$), 99 (no es sap o no es detecta).

2.5.3 Definició d'IMSI i justificació de l'elecció

L'IMSI (*International Mobile Subscriber Identity*) és la identitat internacional de l'abonat a un mòbil. És un codi d'identificació únic integrat en la targeta SIM que permet la identificació en les xarxes GSM i UMTS. D'acord amb l'estàndard ITU E.212 està format per:

- MCC: Codi del país: 3 dígits
- MNC: Codi de la xarxa mòbil: 2 o 3 dígits
- MSIN: Nombre de 9 o 10 dígits com a màxim que conté la identificació de l'estació mòbil.

El nombre màxim de dígits de l'IMSI és 15.

S'ha elegit l'IMSI com identificador de l'equip participant en la cursa Trailwalker, per dues raons:

- És un nombre d'identificació a nivell internacional de la targeta SIM.
- Els dispositius mòbils utilitzats pels participants de la cursa podran ser de molts tipus. La bateria de la majoria és de curta durada. En aquest cas i si no disposen d'una bateria de recanvi, els diferents participants de l'equip necessitaran canviar de dispositiu. És a dir, l'opció d'utilitzar un identificador de dispositiu queda descartada davant aquesta possibilitat.

2.6 Avaluació del consum

2.6.1 Estimació de la intensitat

Es recomana utilitzar una tensió de 7 V a 12 V tal com s'ha explicat en la secció 2.2.2.

Per tal d'estimar la intensitat que circula pel dispositiu s'ha utilitzat una font d'alimentació que ha subministrat una tensió de 9 V. Les intensitats mesurades han estat

Mòduls actius	I (mA)
Només l'Arduino està actiu executant: <code>while (1);</code>	82
S'engega el mòdul 3G/GPRS i l'Arduino executa <code>while (1);</code>	153
S'engega el mòdul 3G/GPRS, el GPS i l'Arduino executa <code>while (1);</code>	203

Taula 2.3: Intensitat depenent dels mòduls actius (cas 9 V).

La placa Arduino disposa d'un regulador de tensió a 5 V. La potència dissipada pel regulador és $(V - 5) \cdot I$, on V és la tensió de la bateria. A 7 V l'eficiència és 5/7 i a 9 V és 5/9. En aquest últim cas s'observa que 4/9 de la potència es malgasta en escalfar el regulador de tensió. Per altra banda, un avantatge per utilitzar 9 V en front de 7 V, és que quan es comenci acabar la bateria, la tensió anirà disminuint. La bateria de 7 V deixarà abans de subministrar la tensió mínima perquè el dispositiu pugui funcionar. A més, com veurem en 2.6.2 l'estalvi en intensitat que aconseguirem reduint de 9 V a 7 V és de només 3 mA quan el mòdul 3G/GPRS està engegat.

2.6.2 Mesures per reduir la intensitat

Apagar el GPS

És necessari apagar el GPS quan no s'està utilitzant ja que suposa una reducció de la intensitat de 50 mA, que representa aproximadament un estalvi del 25%. Aquesta mesura és efectiva i s'aplicarà.

Adormir el microcontrolador

Existeix la possibilitat d'adormir el microcontrolador ATmega328 i despertar-lo després amb una interrupció. En la figura (2.6) es veu com per 5 V i 16 MHz la intensitat és de 9.5 mA. Per tant, amb aquesta mesura s'aconseguiria una reducció de 9.5 mA.

Figure 30-284.ATmega328: Active Supply Current vs. Frequency (1-20MHz)

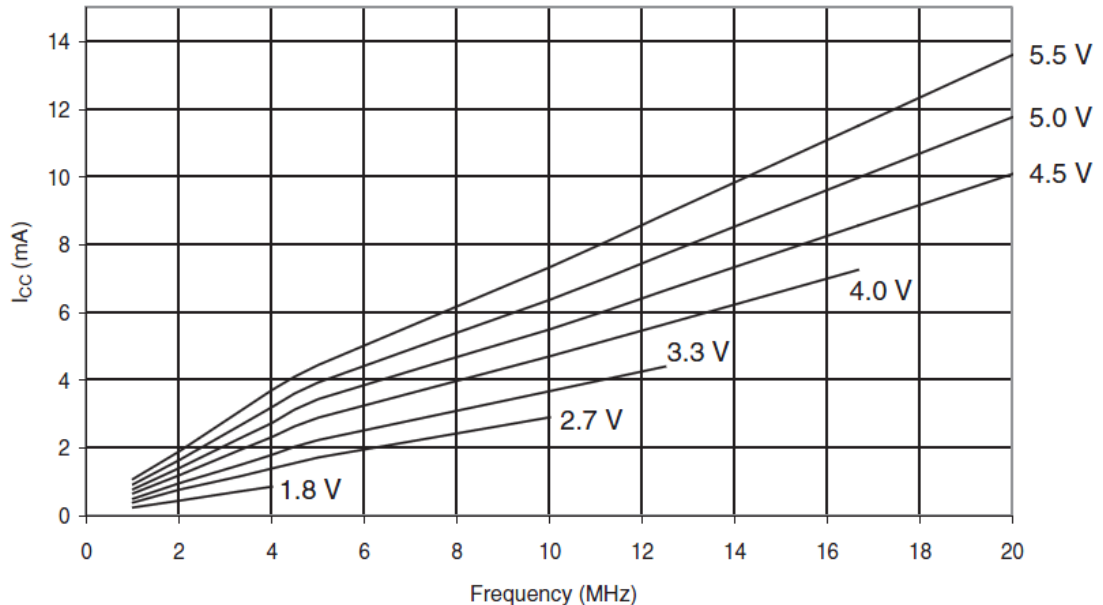


Figura 2.6: Corrent subministrada en funció de la freqüència. [8]

No ha estat possible per manca de temps portar a terme aquesta reducció i es deixa com una línia futura de treball.

Adormir el mòdul 3G/GPRS

Una importat línia futura de treball consistirà en adormir el mòdul 3G/GPRS i despertar-lo quan sigui necessari. A partir de la taula (2.3) es veu que això suposarà una reducció molt significativa d'aproximadament $153 - 82 = 71$ mA.

La comanda AT+CDTRISRMD serveix per configurar la condició per disparar la interrupció per al pin DTR (*Data Terminal Ready*) del mòdul SIM5218 que despertarà el mòdul SIM5218. AT+CDTRISRS permet o no la funció de despertar el mòdul mitjançant el pin DTR. Un pin de propòsit general d'entrada-sortida de la placa Arduino (pins del 0 al 13), que no es faci servir, es pot connectar al pin DTR del mòdul SIM5218. Els passos a executar són els següents:

1. Amb la comanda AT+CDTRISRMD=0,0 s'indica que la interrupció es dispara quan el nivell és baix. El primer 0 indica que es detecta per nivell (1 indicaria per flanc). El segon 0 indica que el nivell és baix (1 indicaria alt).
2. AT+CDTRISRS=1 permet despertar el mòdul quan es desitgi.
3. Posar el pin DTR a nivell baix durant 12 segons: arriba al mòdul SIM5218 la interrupció.

4. Posar el pin DTR a nivell alt: el mòdul SIM5218 ja ha estat despertat. S’ha de tornar el pin DTR a l’estat original, sinó la interrupció seria constantment llançada al mòdul SIM5218 i aquest es col·lapsaria.

La dificultat major es troba en la soldadura que s’ha de realitzar, ja que en la part de l’esquemàtic de la placa 3G/GPRS ([7]) corresponent al mòdul SIM5218 es veu que el pin DTR del mòdul SIM5218 no està connectat a cap pin de la placa 3G/GPRS. En la figura (2.7) es veu aquest fragment de l’esquemàtic.

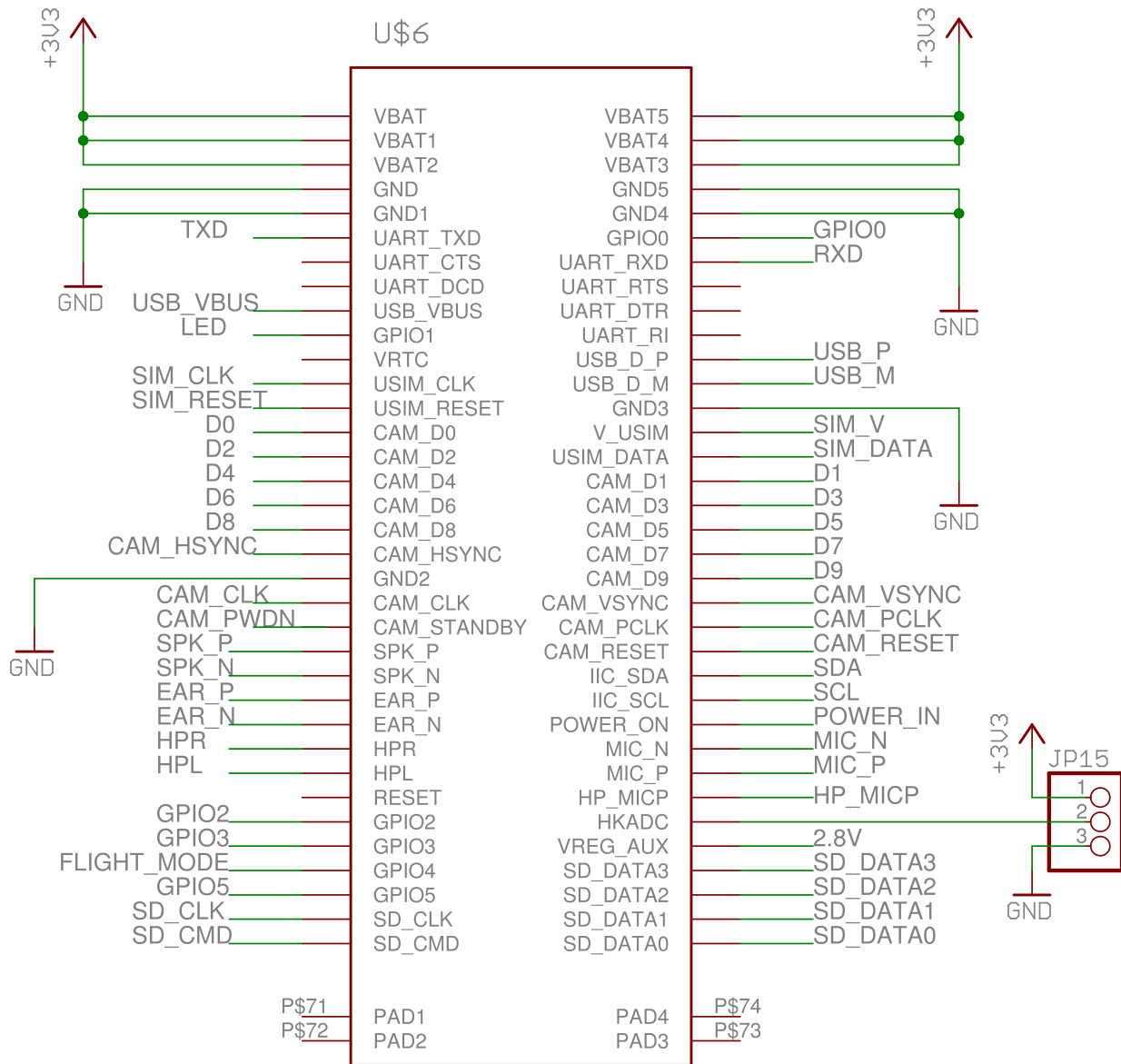


Figura 2.7: Fragment de l’esquemàtic de la placa 3G/GPRS (Mòdul SIM5218)

Disminuir el voltatge de la bateria

Un altra possibilitat per reduir la intensitat és disminuir la tensió. Malgrat ser la tensió mínima de 6 V, aquest valor no es recomana perquè la placa podria esdevenir inestable. De totes maneres, aquest valor serveix per esbrinar el màxim estalvi possible.

- Per exemple si la tensió utilitzada és de 6 V i en el cas que només la placa Arduino estigui activa (cas en què s'adorm el mòdul SIM5218), la intensitat és reduïda a 71 mA. Tenint en compte que a 9 V era de 82 mA, en el cas de disminuir el voltatge de la bateria i adormir el mòdul, la reducció seria inferior a 11 mA.
- En el cas en què el mòdul 3G/GPRS estigui actiu i el GPS apagat, la intensitat es reduïria a 150 mA. En aquest cas, la diferència no és gaire significativa, és només de 3 mA.

La conclusió és que en el cas de poder adormir el mòdul, seria millor reduir una mica el voltatge, però sempre intentant que sigui prou superior a 7 V per tal que els efectes del desgast de la bateria no es notin massa ràpidament. En el cas de no poder adormir el mòdul, es pot treballar perfectament amb 9 V.

Apagar el mòdul

Aquesta és una opció molt dràstica. Amb la comanda AT+CPOF, es pot apagar el mòdul. Després es torna a engegar amb el mateix procediment utilitzat per engegar-lo per primera vegada. Per exemple, caldrà tornar a introduir el PIN amb la comanda AT+CPIN. El desavantatge d'aquest mètode és que aleshores és més costós trobar quines són les coordenades. El TTFF (*Time To First Fix*) pot arribar a ser molt llarg, a vegades, inclús de minuts. Quan el receptor GPS s'engega per primera vegada, els satèl·lits han de transmetre dades de forma contínua durant 40 segons i sense interrupció al receptor GPS. Aquest temps pot ser molt més llarg si el senyal és dèbil i la transmissió s'interromp constantment, com podria ocórrer en un bosc molt espès. Això implicaria un major consum d'energia.

Altres

S'han realitzat altres proves per veure si s'aconsegueix una reducció de la intensitat, però no hi ha hagut èxit:

- La funció `delay()` no redueix la intensitat. Tot i que aquesta funció provoca que gairebé tota l'activitat s'aturi, la intensitat observada ha estat 1 mA major. És a dir, el valor obtingut ha estat de 83 mA, en front dels 82 mA obtinguts amb l'execució de `while(1)`;
- La comanda AT+CFUN=0 no redueix tampoc la intensitat. El nivell 0 és el nivell on es consumeix el mínim nivell d'energia. En les proves realitzades només s'ha apreciat una reducció d'1 mA, amb l'inconvenient que hi ha hagut molta variabilitat en les diferents mesures preses de la intensitat respecte el valor de 152 mA, a on semblava tendir. Per tant, l'ús d'aquesta comanda s'ha descartat.

2.6.3 Estimació de la intensitat mitjana i del tipus de bateria

Apagant el GPS

Tenir en compte les consideracions de 2.6.1 i sota el supòsit que la tensió d'entrada és de 9 V, es realitzarà una estimació de l'intensitat mitjana.

La intensitat mitjana en aquest període de 615 segons és

$$I = \frac{400 \cdot 4.5 + 153 \cdot 6 + 500 \cdot 4.5 + 153 \cdot 600}{4.5 + 6 + 4.5 + 600} = 157.35 \text{ mA}$$

Interval de temps en què	I (mA)	Temps (segons)
El GPS està actiu i adquirint les dades	400	4.5
L'Arduino prepara les dades que s'hauran de transmetre al servidor	153	6
S'envien les dades al servidor	500	4.5
Espera de 10 minuts	153	600
Globlament	157.35	615

Taula 2.4: Avaluació de la intensitat mitjana.

Per tant

$$157.35 \text{ mA} \cdot 32 \text{ h} = 5035.2 \text{ mA} \cdot \text{h}$$

Caldria buscar una bateria de 5000 mA · h amb un voltatge de 7.5 a 9 V.

Adormint el microcontrolador i el mòdul

En el cas que d'adormir el microcontrolador de la placa Arduino i i el mòdul SIM5218 de la placa 3G/GPRS, s'obtidria un estalvi d'intensitat de $9.5 + 71 = 80.5$ mA, durant l'estona d'espera de 10 minuts, és a dir, en lloc de 153 mA la intensitat seria de 72.5 mA. Per tant, la intensitat mitjana resultaria de

$$I = \frac{400 \cdot 4.5 + 153 \cdot 6 + 500 \cdot 4.5 + 72.5 \cdot 600}{4.5 + 6 + 4.5 + 600} = 78.81 \text{ mA}$$

Per tant

$$78.81 \text{ mA} \cdot 32 \text{ h} = 2521.92 \text{ mA} \cdot \text{h}$$

En aquest cas es podrien utilitzar, en sèrie, 5 o 6 piles alcalines del tipus AA. Aquestes piles ofereixen 2700 a 2900 mA · h i un voltatge d'1.5 V.

Nombre de piles AA en sèrie	V	mA · h
5	7.5	2700-2900
6	9	2700-2900

Taula 2.5: Piles AA necessàries en el cas d'adormir el microcontrolador i el mòdul.

Apagant el mòdul

En aquest cas i suposant que el TTFF és de 40 segons, la intensitat mitjana resultaria de

$$I = \frac{400 \cdot 40 + 153 \cdot 6 + 500 \cdot 4.5 + 82 \cdot 600}{40 + 6 + 4.5 + 600} = 105.10 \text{ mA}$$

Per tant,

$$105.10 \text{ mA} \cdot 32 \text{ h} = 3363.2 \text{ mA} \cdot \text{h}$$

Si s'utilitzessin 5 o 6 piles alcalines del tipus AA, la capacitat de corrent seria de 2700 a 2900 mA · h.

Per tant, com que

$$2900/3363.2 = 86.2\%$$

només es podria cobrir el 86.2% de les 32 hores, que és el temps màxim de la cursa, és a dir, 27 hores i 35 minuts. Tenint en compte que la cursa es sol acabar en unes 24 hores, podria ser suficient.

Apagant el mòdul i enviant dades cada 20 minuts

En el cas, de realitzar esperes de 20 minuts en lloc de 10 minuts

$$I = \frac{400 \cdot 40 + 153 \cdot 6 + 500 \cdot 4.5 + 82 \cdot 1200}{40 + 6 + 4.5 + 1200} = 94.02 \text{ mA}$$

Per tant,

$$94.02 \text{ mA} \cdot 32 \text{ h} = 3008.64 \text{ mA} \cdot \text{h}$$

En aquest cas es podria arribar a cobrir amb 5 o 6 piles alcalines AA de 2900 mA · h en sèrie, 30 hores i 51 minuts de la cursa.

Apagant el mòdul, enviant dades cada 20 minuts i adormint el microcontrolador

Si a més a més s'adorm el microcontrolador

$$I = \frac{400 \cdot 40 + 153 \cdot 6 + 500 \cdot 4.5 + 71 \cdot 1200}{40 + 6 + 4.5 + 1200} = 83.46 \text{ mA}$$

Per tant,

$$83.46 \text{ mA} \cdot 32 \text{ h} = 2670.72 \text{ mA} \cdot \text{h}$$

En aquest cas és suficient 5 o 6 piles alcalines en sèrie per cobrir les 32 hores de la cursa.

Apagant el mòdul, enviant dades cada 10 minuts i adormint el microcontrolador

En el cas d'adormir el microcontrolador i fer els enviaments cada 10 minuts

$$I = \frac{400 \cdot 40 + 153 \cdot 6 + 500 \cdot 4.5 + 71 \cdot 600}{40 + 6 + 4.5 + 600} = 94.95 \text{ mA}$$

Per tant,

$$94.95 \text{ mA} \cdot 32 \text{ h} = 3038.40 \text{ mA} \cdot \text{h}$$

En aquest cas es podria arribar a cobrir amb 5 o 6 piles alcalines AA de 2900 mA · h en sèrie, 30 hores i 32 minuts de la cursa.

2.6.4 Estimació de la intensitat mitjana mitjançant una pila HR22

Aquest conjunt de proves s'han portat a terme per confirmar les estimacions obtingudes per la intensitat mitjana. Per realitzar les estimacions anteriors, s'ha suposat, per exemple, una intensitat constant de 400 mA durant tot l'interval de temps en què el GPS està adquirint dades. Aquest supòsit és òbviament fals i es tracta d'una simplificació. Les mesures observades en el visor de la font d'alimentació variaven durant aquest interval, fins i tot, a vegades apareixien pics molt més alts. Per ser exactes, es tractaria de calcular la intensitat mitjana I_m mitjançant

$$I_m = \frac{\lim_{t \rightarrow \infty} \int_0^t I(s) ds}{t}$$

Tot i així, les tècniques que es requeririen s'escapen de l'abast d'aquest treball i es buscarà una tècnica alternativa.

Es realitzaran dues proves: una amb el mòdul SIM5218 tota l'estona engegat i l'altra amb aquest mòdul apagat en els 10 minuts d'espera.

El mòdul SIM5218 engegat

S'ha utilitzat un pila HR22 recarregable (170 mA·h, 9 V) i un adaptador amb un connector de 21 mm centre-positiu. S'ha aconseguit enviar dades durant 45 minuts. En acabar i mesurar la pila s'ha obtingut una tensió de 8.1 V. Aquesta tensió s'ha recuperat i al cap d'una estona ha arribat a 9 V. Però la pila ja no ha estat ni tan sols capaç d'engegar el mòdul novament.

L'explicació és molt senzilla: una pila es pot simular com una font d'alimentació i una resistència en sèrie. A mida que la pila es va gastant, la font d'alimentació dona menys voltatge i la resistència va augmentant. Aquest voltatge ha d'alimentar el dispositiu mòbil i la resistència.

El dispositiu mòbil pot tenir pics d'intensitat de fins a 2 A, segons les especificacions elèctriques de la placa 3G/GPRS en [7]. Tot i que, utilitzant la font d'alimentació, no s'han observat pics tan alts, aquests pics es poden donar. Durant un d'aquests pics, el diferencial de tensió corresponent a la resistència és molt alt, tant que la tensió que queda pel dispositiu mòbil no és suficient.

Aquest pics es produeixen en enviar comandes AT al mòdul. Es detecta que la bateria ha arribat a aquest estat perquè el led groc de la placa Arduino, queda encès i no s'apaga. L'aplicació està programada per tal que quan s'envii una comanda AT, el led s'encengui en espera de resposta i s'apagui si ja no s'espera cap resposta perquè ja s'ha rebut. Per tant que el led groc quedi encès indica que s'ha donat una de les següents opcions:

- No s'ha subministrat a la placa intensitat suficient. Aquest cas també s'ha observat quan es van realitzar les proves amb la font d'alimentació i es va limitar la intensitat a valors inferiors a un cert llindar insuficient, com 300 mA.
- La tensió subministrada no és suficient.
- El mòdul no està encès. Aquest cas és un cas evident, ja que aleshores el mòdul no enviarà cap resposta a cap comanda AT que se li pugui enviar. Aquest cas es dona, quan s'està intentant alimentar el dispositiu amb una pila una mica gastada. La placa 3G/GPRS ni tan sols s'engegarà i el led groc quedarà encès en enviar una comanda, ja que el mòdul apagat no enviarà cap resposta.

Fent un petit càlcul s'obté que la capacitat de corrent de la bateria per ser suficient per les 32 hores de cursa és de

$$170 \text{ mA} \cdot \text{h} \cdot \frac{32 \text{ h}}{0.75 \text{ h}} = 7253.3 \text{ mA} \cdot \text{h}$$

Aquest valor és molt superior a l'estimació de 5035.2 mA·h realitzada en 2.6.3.

No és contradictori. El que passaria amb una bateria d'aproximadament 5000 mA·h, és que deixaria de ser útil pel dispositiu mòbil abans d'haver esgotat tota la seva capacitat de corrent. La causa serien aquests pics de corrent tan alts. Per tant, hi ha dues possibilitats:

- Augmentar la capacitat de corrent de la bateria i agafar-la al voltant de 7200 mA·h.
- Incrementar el voltatge de la bateria al màxim possible recomanat, és a dir, 12 V.

Només cal afegir, que la mesura proposada anteriorment de reduir el voltatge de la bateria per tal de reduir la intensitat, tot i que és certa, és completament inadequada, tal com s'acaba de veure.

El mòdul SIM5218 apagat durant l'espera de 10 minuts

S'ha utilitzat la mateixa pila HR22 recarregable (170 mA·h, 9 V), novament recarregada. S'utilitza per carregar les piles un carregador que detecta quan la pila està carregada. És a dir, la pila d'aquest experiment i de l'anterior té les mateixes característiques. En aquest cas, la pila ha durat aproximadament 2 hores 15 minuts. Es detecta que la pila resulta insuficient perquè s'encén el llum groc. L'hora de l'última dada es pot esbrinar fàcilment amb una operació *query* (<http://napols.uoc.es/arcgis/rest/services/TrailWalker/MapServer/0/query>) En aquest cas la capacitat total de la bateria hauria de ser de

$$170 \text{ mA} \cdot \text{h} \cdot \frac{32 \text{ h}}{2.25 \text{ h}} = 2417.8 \text{ mA} \cdot \text{h}$$

És a dir, es podran utilitzar piles alcalines AA de 2900 mA·h en sèrie. Tenint en compte les consideracions de l'apartat anterior i que la tensió màxima recomanada per la placa Arduino és de 12 V, s'utilitzaran 8 piles d'aquest tipus en sèrie. El voltatge total serà de 12 V i la capacitat de corrent total de 2900 mA·h.

S'ha observat una diferència molt gran entre la durada de la bateria apagant i sense apagar el mòdul SIM5218. Caldria repetir els dos experiments i calcular la durada mitjana de la bateria en cadascun dels casos. Els pics d'intensitat no són constants. Els 45 minuts del primer experiment podrien haver estat causats per un pic d'intensitat desafortunadament alt. Les 2 hores i 15 minuts es podrien haver assolit en haver tingut la sort de no haver-se produït un pic exageradament alt. Caldria mesurar els efectes, en la durada de la bateria, de l'aleatorietat en els pics d'intensitat.

Capítol 3

Serveis Web

Un servei *Web* és un mètode de comunicació de dos dispositius electrònics en la *World Wide Web*, és una funció de programari proporcionada en una adreça de xarxa de la *Web*. Aquest sistema de programari dissenyat per suportar una interacció màquina a màquina sobre una xarxa té les següents característiques:

- És accessible a través de la *Web*. No requereix cap infraestructura ni topologia de xarxa especial. Es pot aprofitar la infraestructura *Web* ja existent.
- Proporciona una interfície ben definida. La implementació real queda oculta al consumidor del servei.
- El proveïdor del servei i el consumidor poden utilitzar tecnologies diferents per poder interactuar. Hi ha serveis *Web* destinats a ser consumits per clients de la mateixa organització, d'altres organitzacions o bé per qualsevol client de tot el món. Com més interoperabilitat es desitgi aconseguir menys acoblament hi ha d'haver entre el servidor i el client.

A grans trets, es pot parlar de dos classes de serveis *Web*: els serveis *Web* REST, en els quals es manipulen representacions dels recursos *Web* utilitzant un conjunt uniforme d'operacions sense estat (*stateless*) i els serveis *Web* arbitraris que poden disposar d'un conjunt arbitrari d'operacions.

3.1 SOAP

SOAP (*Simple Object Access Protocol*) és una especificació de protocols per l'intercanvi d'informació estructurada en la implementació de serveis *Web* en xarxes. Es recolza en l'XML (*Extensible Markup Language*) pel format dels seus missatges i en altres protocols de la capa d'aplicació, com HTTP (*Hypertext Transfer Protocol*) i SMTP (*Simple Mail Transfer Protocol*), per la negociació i transmissió de missatges.

- XML ha estat elegit com el format estàndard dels missatges perquè és un format molt estès, de codi obert i amb moltes eines gratuïtes disponibles. La longitud de la sintaxi de XML és en part beneficiosa ja que facilita la lectura per les persones i la detecció d'errors, però a la vegada alenteix la velocitat de processament.
- HTTP i SMTP són protocols a nivell d'aplicació aptes pel transport de SOAP. De totes maneres HTTP ha guanyat en acceptació ja que funciona en la infraestructura actual d'Internet. SOAP també pot ser utilitzat sobre HTTPS, que proporciona seguretat sobre el servei *Web*.

SOAP es pot considerar la capa base per la pila de protocols de serveis *Web* ja que proporciona una marc pel pas de missatges sobre el qual es poden construir els serveis *Web*. Aquest protocol consisteix en tres parts: un sobre (*envelope*) que defineix que hi ha en el missatge i com es processa, un conjunt de regles de codificació per expressar instàncies del tipus de dades definides per l'aplicació i un conveni per representar crides i respostes a funcions. Per exemple, amb l'objectiu de realitzar una consulta es podria enviar un missatge SOAP a una URI amb serveis *Web* disponibles tals com una base de dades. La resposta consistiria en un document XML amb les dades demanades.

L'arquitectura SOAP està formada per diferents capes d'especificacions: format dels missatges, *Message Exchange Patterns* (MEP), enllaços del protocol de transport subjacents, models de processament de missatges i extensibilitat del protocol. SOAP és el successor de XML-RPC (*Extensible Markup Language-Remote Procedure Call*).

Els resultats de les primeres versions de SOAP a nivell d'interoperabilitat van ser molt dolents. La complexitat de l'estàndard va provocar que les implementacions dels diferents fabricants fossin incompatibles. Per tal de solucionar-ho es va treure un altre estàndard, WS-I Basic Profile, en el qual es recomana utilitzar una versió restringida de SOAP i, a més, s'aclareixen certs aspectes de WSDL. Amb el temps s'ha aconseguit una interoperabilitat acceptable.

Conscients d'aquests problemes i amb la necessitat d'interoperabilitat global s'ha impulsat un nou, en realitat vell, enfocament per la construcció de serveis *Web*. Aquest enfocament és REST. S'utilitza en les empreses més grans del món a nivell global. En la última versió de WSDL, WSDL2.0, s'han realitzat simplificacions com l'eliminació dels missatges. A més, no tan sols suporta SOAP, sinó que també suporta REST. De totes maneres, tot i ser possible, els serveis REST no necessiten utilitzar WSDL2.0.

3.2 REST

REST (*Representational State Transfer*) és una tècnica d'arquitectura *software* per a sistemes hipermitjà distribuïts com la *World Wide Web*. REST s'ha transformat en el model de disseny de serveis *Web* predominant. Qualsevol arquitectura de serveis distribuïts que compleixi les següents característiques, es pot considerar com una arquitectura REST:

- No es publiquen serveis RPC (*Remote Procedure Call*). El que es publica són recursos. Un recurs es pot considerar com una entitat a la qual es pot accedir públicament. Cada recurs té un identificador global diferent. Cada recurs té un estat intern al qual no es pot accedir des de l'exterior. El que sí que és accessible des de l'exterior és una o varies representacions d'aquest estat. Per representació s'entén el format de dades que s'utilitza per la transferència, entre el client i el servidor, d'una còpia de l'estat públic del recurs. La implementació del recurs decideix quina informació és visible des de l'exterior i quines representacions es suporten. Un representació podria ser un document HTML, un document XML, un JSON, un PNG, etc.
- No es poden definir operacions arbitràries sobre un recursos. Tots els recursos comparteixen una interfície única i constant. Tots els recursos tenen les mateixes operacions:
 - CREATE. El client envia al servidor una petició per crear un nou recurs. Opcionalment el client pot enviar una representació de l'estat inicial d'aquest recurs. El servidor respon amb un identificador global d'aquest recurs.

- DELETE. El client elimina un recurs del servidor. El client ha de conèixer l'identificador d'aquest recurs.
- READ. El client pot llegir una representació de l'estat del recurs, identificat amb el seu identificador global.
- UPDATE. S'actualitza en el servidor l'estat d'un recurs amb les modificacions realitzades pel client.

La implementació del servei és lliure de prohibir alguns d'aquests mètodes per un recurs concret. Cal definir el model de dades que es publicaran i les representacions que es suporten.

- En realitzar un UPDATE, es realitzen les validacions i actualitzacions necessàries en altres recursos per mantenir una consistència global del sistema. Els recursos mantenen interrelacions. La topologia d'aquestes interrelacions pot arribar a ser complexa.

REST és un model de disseny pels serveis *Web* amb un enfocament madur. Un exemple clar es troba en la *World Wide Web*. En efecte:

- Cada pàgina *Web* és un recurs.
- Cada recurs té un identificador únic, que és l'URI.
- Donat un recurs, mitjançant el protocol HTTP s'opera sobre aquests recursos. Mitjançant camps de la capçalera com Accept o Content-Type s'especifiquen les representacions que entenen el servidor i el client i quina representació s'utilitza en el paquet concret que transporta l'estat del recurs.
- GET, DELETE, PUT i POST es corresponen respectivament amb READ, DELETE, UPDATE I CREATE.
- Les representacions utilitzades s'especifiquen mitjançant tipus MIME, la majoria dels quals són estàndard, com XML o JSON. L'ús de tipus estàndard en facilita la interoperabilitat.
- L'aplicació client típica és el navegador, que realitza peticions GET sobre les URI corresponents per tal d'obtenir representacions de les diferents pàgines *Web*. Habitualment una pàgina *Web* admet una sola representació, HTML, PDF, text pla, imatge. Però una URI pot suportar també diferents representacions.

La *Web* és un exemple perfecte de serveis distribuïts a nivell global gairebé totalment interoperables. La *Web* està formada per serveis REST que en general només suporten HTML. Gairebé qualsevol pàgina HTML és interoperable amb qualsevol navegador, i gairebé qualsevol pàgina HTML pot interoperar amb qualsevol altra pàgina realitzada per un tercer.

3.3 Avantatges de REST

- No cal que el servidor mantingui activa la sessió. El protocol funciona amb un simple diàleg de petició i resposta.
- Es pot utilitzar HTTP, URIs i tipus MIME per publicar serveis de dades conformes als principis REST. La possibilitat de múltiples representacions o formats de dades existeix de manera natural. No cal limitar-se a XML. El protocol HTTP ja el suporta de manera natural.

- HTTP és un protocol que segueix els principis REST. Els serveis *Web* aprofiten tota la infraestructura de la xarxa ja existent: memòria cau, *proxies*, tallafocs, compressió, etc. Amb REST no es crea un nou un protocol i s'encaixa dins d'HTTP, sinó que s'utilitza el propi HTTP. En canvi, amb SOAP es poden invocar els procediments remots a través de diferents protocols: SMTP, HTTP, etc. Aquesta neutralitat respecte la infraestructura *Web* evita que pugui aprofitar els avantatges d'aquesta infraestructura i, fins i tot, pot provocar problemes.
- Les eines són molt senzilles, no és ni necessari entorn de treball per sistemes senzills. No es generen quantitats desmesurades de codi. Un mòbil, per exemple, té la capacitat suficient per actuar com un client REST. En canvi, el format SOAP no és concís, diu poc amb moltes paraules i això és un desavantatge quan es treballa amb amples de banda reduïts. SOAP consumeix molts recursos. Per tant, és difícil que sigui utilitzat per clients amb dispositius mòbils.
- El codi és fàcil de mantenir. En canvi, la informació en WSDL és tan gran i tan complexa que genera un gran acoblament entre el consumidor i el proveïdor del servei. Durant el manteniment, quan sorgeix un problema pot arribar a ser molt difícil depurar un error en un servei *Web* basat en SOAP i esbrinar el que està fent el sistema i el que està ocorrent.

Capítol 4

ArcGIS

4.1 Introducció

ArcGIS és un sistema complet que cobreix tot el cicle de producció d'un mapa. Permet recollir, organitzar, gestionar, analitzar, comunicar i distribuir informació geogràfica. És una plataforma capdavantera a nivell mundial per construir i utilitzar sistemes d'informació geogràfica (SIG). ArcGIS permet que la informació es pugui publicar, de tal manera que tothom hi pugui accedir. El sistema inclou *software*, una infraestructura basada en Internet, eines professionals, recursos configurables tals com mapes base, plantilles, etc. ArcGIS permet:

4.1.1 Crear mapes intel·ligents

Es pot crear una gran varietat de mapes: mapes *Web* accessibles des dels navegadors o els dispositius mòbils, composicions de mapes impreses en gran format, mapes inclosos en informes i presentacions, llibres de mapes, atlas, mapes incrustats en aplicacions,... Aquests mapes integren diferents capes d'informació geogràfica i descriptiva:

- Un mapa comença amb un mapa base. ArcGIS té incorporat un conjunt de mapes base que inclou: topografia, imatges, carrers, terrenys, oceans, etc. Hi ha molts més mapes especialitzats com hidrologia, ús de les terres i geologia. Fins i tot es poden crear mapes bases.
- Després s'afegeixen capes de dades operatives, s'escull la simbologia, etiquetes, rangs d'escala i es configuren finestres emergents per mostrar els atributs clau.
- Segons el propòsit del mapa es poden configurar eines addicionals, tals com eines d'edició, accés a models analítics, barra de desplaçament en el temps.

4.1.2 Compilar informació geogràfica

Es poden sintetitzar dades provinents de múltiples fonts en una única vista. Aquestes fonts poden ser bases de dades geogràfiques, fitxers, fulls de càlcul, fotografies i vídeos etiquetats amb la ubicació geogràfica, dades CAD, imatges directes des de sensors, imatges aèries i per satèl·lit, etc. De fet, tota informació amb un referència geogràfica tal com nom de carrer, ciutat, identificador de parcel·la, coordenades GPS, etc, pot ser ubicada en un mapa.

ArcGIS també permet crear fàcilment dades geogràfiques i emmagatzemar-les en la base de dades geogràfiques del sistema, mitjançant la tecnologia mòbil o *Web* i aplicacions de recollida de dades.

Això permet a les organitzacions recollir en el seu SIG dades recents i respondre ràpidament a qualsevol situació.

4.1.3 Crear i gestionar bases de dades geogràfiques

Les bases de dades geogràfiques es troben en el cor d'un treball SIG professional. Una base de dades geogràfica permet emmagatzemar la informació de manera estructurada per tal de gestionar-la, actualitzar-la, reutilitzar-la i compartir-la fàcilment. Les capes de dades que es gestionen i s'emmagatzemen poden ser parcel·les, fronteres administratives, hidrografia, altituds, etc.

Les dades poden ser editades i utilitzades per múltiples usuaris. Els canvis seran immediatament visibles per a tots els usuaris. Les dades multiusuari estan implementades per sistemes de bases de dades relacionals empresarials estàndard, tals com Oracle, SQL Server, PostgreSQL, Informix i DB2.

ArcGIS suporta moltes possibilitats en el flux d'informació. En situacions en les quals no es pot connectar amb la base de dades, ArcGIS *Mobile* permet a un dispositiu mòbil fer edicions i enviar-les després. Diferents rèpliques de bases de dades geogràfiques poden sincronitzar-se. Es poden crear arxius històrics on es vegi el canvi de les dades al llarg del temps.

4.1.4 Resoldre problemes amb anàlisi espacial

L'objectiu es extreure nova informació de les dades per prendre millors decisions. La simbologia inherent en els mapes convida de manera natural a l'anàlisi i interpretació de patrons i relacions. ArcGIS, a més a més, utilitza eines matemàtiques, estadístiques i geogràfiques per resoldre una infinitat de problemes: trobar característiques que coincideixen amb un cert criteri, modelar processos naturals com el flux de l'aigua sobre un terreny o utilitzar estadística espacial per esbrinar la distribució d'uns punts. D'aquesta manera es poden descobrir relacions i patrons que d'una altra manera romandrien ocults. A més, es pot proporcionar accés als models analítics per tal que tothom els pugui utilitzar. Es poden crear models i *scripts* per tal d'automatitzar processos i després publicar-los i compartir-los.

4.1.5 Crear aplicacions basades en mapes

La creació d'aplicacions permet convertir els mapes, dades i eines en productes que tothom pot utilitzar en un ampli ventall de situacions. Les aplicacions poden ser aplicacions *Web* o aplicacions per ordinadors personals, *smartphones*, *tablets* o altres dispositius mòbils. Poden ser aplicacions dirigides a un públic general o focalitzades en tasques, com en el cas d'aplicacions de camp per a enginyers.

No és necessari ser un desenvolupador per crear aplicacions. Mitjançant l'ArcGIS Online es poden crear aplicacions *Web*, utilitzant les seves plantilles genèriques, que a més són configurables. En la comunitat d'usuaris d'ArcGIS Online es poden trobar també moltes plantilles configurables per aplicacions genèriques creades amb Javascript.

4.1.6 Comunicar i compartir informació

Un ús molt rellevant de qualsevol SIG és com a eina de comunicació, sense necessitat que l'audiència sigui experta en SIG. Una vegada el mapa ha estat creat es pot compartir. Els mapes ArcGIS es

poden publicar a Internet i són fàcils d'utilitzar. Es poden realitzar presentacions amb diapositives dinàmiques, que permeten interactuar amb el mapa.

4.2 Mòduls d'ArcGIS

4.2.1 ArcGIS Desktop

És el producte bàsic orientat a professionals en aplicacions SIG per compilar, analitzar, gestionar informació geogràfica i compartir-la. Permet construir dades, mapes, models i aplicacions. Permet compartir paquets i pujar serveis als servidors ArcGIS.

ArcGIS Desktop es troba disponible en tres nivells de funcionalitats diferents:

- Bàsic: focalitzat en l'ús de dades, mapes i anàlisis.
- Estàndard: afegeix l'edició avançada de bases de dades geogràfiques i la creació de dades.
- Avançada: Inclou totes les funcionalitats SIG i moltes eines de geoprocessament.

Existeixen extensions especialitzades: ArcGIS 3D Analyst, ArcGIS Business Analyst, etc.

4.2.2 ArcGIS Server

Agafa els recursos utilitzats en ArcGIS per a escriptori i els exposa com serveis *Web* que poden ser utilitzats en un ampli ventall de dispositius i aplicacions. Aquests serveis poden ser de: mapa, imatge, característiques, geoprocessament, geocodificació, geometria, etc. Per exemple, a partir d'un document ArcMap es pot crear un servei de mapa. Es pot compartir aquest servei a través de protocols *Web* comuns. L'accés i la integració és senzilla utilitzant per exemple, REST, KML, etc. Aquests serveis es poden utilitzar en aplicacions d'escriptori, navegadors *Web*, mòbils i *tablets*.

Utilitza interfícies estàndard per tots els serveis: REST amb *GeoServices REST Specification*, SOAP XML, KML, OGC (incloent les especificacions WMS, WCS, WFS, WPS i WMTS), SQL. Suporta diferents implementacions SIG: empresa, grup de treball, etc. Suporta múltiples usuaris i proporciona accés a una extensa funcionalitat SIG. Inclou un SIG ampli, serveis de mapes, interfícies de serveis *Web* obertes i basades en estàndards, una consola administrativa per la configuració i gestió del servidor, diferents sistemes de gestió de bases de dades geogràfiques empresarials (Oracle, SQL Server, SQL Server Express, DB2, Informix, PostgreSQL, Netezza). Permet la gestió de bases de dades multiusuari. Es poden sincronitzar les diferents rèpliques de la base de dades. Suporta un nombre molt gran d'usuaris, fluxos intensos de dades, transaccions llargues. Gaudeix de tots els beneficis de les bases de dades relacionals: integritat, seguretat, còpies de seguretat, etc. També permet connectar-se a base de dades externes. El servidor pot executar-se en entorns Windows i Linux.

4.2.3 ArcGIS Online

És un lloc que permet transformar els propis mapes en mapes accessibles a través de la *Web* i poder-los compartir. Disposa de múltiples aplicacions i un gran ventall d'opcions depenent de quins siguin els usuaris finals.

És utilitzat també per professionals SIG per gestionar i publicar la seva informació. Per altra banda també permet que els que no siguin experts en SIG puguin confeccionar els seus propis

mapes utilitzant capes ja existents amb altres capes obtingudes a partir de fulls de càlculs o, fins i tot, dibuixant literalment en el mapa mateix.

Es pot accedir anònimament per buscar i veure mapes publicats per altres usuaris i compartits amb tothom. Per crear els propis mapes i compartir-los és necessari crear un compte en ArcGIS Online, que proporciona un espai per emmagatzemar els mapes i la informació que es vulgui compartir. Per l'ús més avançat que pot dur a terme una organització, serà necessari realitzar un subscripció per a organitzacions. L'organització ja no necessitarà un servidor propi per publicar, ja que això es podrà realitzar en el núvol ArcGIS Online.

4.3 Servei REST

El servidor ArcGIS permet publicar serveis per la visualització, l'anàlisi espacial i la gestió de dades espacials. Una vegada els serveis han estat publicats, diferents marcs de treball estan disponibles per accedir i utilitzar aquests serveis, incloent l'API de JavaScript per a ArcGIS.

4.3.1 El servei de directoris i l'API REST

Per obrir el directori de serveis d'una màquina amb el servidor ArcGIS, cal conèixer l'URL de la instància del servidor. En una instal·lació típica d'un servidor ArcGIS, s'instal·la una instància anomenada *arcgis* en l'arrel del servidor *Web*. El directori de serveis es troba en la carpeta *rest* dins aquesta instància. El directori de serveis que s'utilitza en aquest cas és <http://napols.uoc.es/arcgis/rest/services> i els serveis són:

- <http://napols.uoc.es/arcgis/rest/services/TrailWalker/MapServer>
- <http://napols.uoc.es/arcgis/rest/services/rutes/MapServer>
- <http://napols.uoc.es/arcgis/rest/services/TrailWalker/FeatureServer>
- <http://napols.uoc.es/arcgis/rest/services/UltimaEntrada/MapServer>

El tipus de servei s'adjunta al final de l'URL: MapServer, FeatureServer,... També es poden identificar característiques en capes particulars. L'identificador de la capa és un enter.

- <http://napols.uoc.es/arcgis/rest/services/TrailWalker/MapServer/0>: capa corresponent als punts de seguiment enviats pels diferents equips participants.
- <http://napols.uoc.es/arcgis/rest/services/rutes/MapServer/0>: capa corresponent als punts de control del TrailWalker.
- <http://napols.uoc.es/arcgis/rest/services/rutes/MapServer/1>: capa corresponent a les etapes del TrailWalker.
- <http://napols.uoc.es/arcgis/rest/services/TrailWalker/FeatureServer/0>: capa corresponent als punts de seguiment enviats pels diferents equips participants. Aquesta capa suporta l'operació *Add Feature*. Si el servei fos MapServer en lloc de FeatureServer aquesta capa no suportaria aquesta operació.

Les capes es llisten en la pàgina de propietats del servei de mapa. En el cas de necessitar realitzar una consulta, es pot buscar prèviament, en les propietats de la capa, la llista de camps existents. Les carpetes i serveis poden ser segurs, de manera que només usuaris autoritzats puguin accedir-hi. L'API REST del servidor ArcGIS proporciona una manera senzilla d'utilitzar els serveis proporcionats pel servidor ArcGIS mitjançant el directori de serveis. REST permet l'ús de recursos *Web* mitjançant l'encapsulament d'informació en l'URL i la cadena de consulta de la petició.

L'ArcGIS Server inclou un API REST per accedir als serveis. L'API REST suporta la major part de la funcionalitat disponible en l'API SOAP de l'ArcGIS Server. L'API REST permet la creació d'aplicacions *Web* que estiguin estrictament al costat del client.

L'API REST retorna dades en *JavaScript Object Notation* (JSON). Generalment no és necessari treballar amb dades en format JSON. JavaScript permet la manipulació d'aquest tipus de dades. Les dades JSON estan aparellades amb el format {nom : valor}. El valor pot ser de molts tipus, incloent una cadena de caràcters, un nombre o un vector. Els diferents elements d'un vector van entre [] i es separen entre comes. Un únic parell ja és un objecte JSON. Un objecte JSON també pot estar constituït per múltiples parells.

Es pot referenciar a objectes JSON directament utilitzant JavaScript. Cal conèixer els noms dels objectes dins l'objecte JSON. Aquests noms es poden obtenir examinant el directori de serveis.

4.4 Integració del dispositiu mòbil amb el servei REST

4.4.1 Formularis

L'enviament de dades d'un formulari es pot fer mitjançant els mètodes HTTP: GET i POST. Amb el mètode POST les dades s'inclouen en el cos del formulari. S'utilitza aquest mètode quan es necessita modificar una base de dades, realitzar una subscripció a un servei, etc. En canvi si l'enviament de dades no implica cap modificació s'utilitza el mètode GET. En aquest cas, les dades s'afegeixen a l'URI amb un ? com a separador.

El mètode GET es restringeix a dades amb caràcters ASCII. En canvi amb el mètode POST es pot arribar a cobrir el joc de caràcters sencer ISO10646. Amb el mètode GET el contingut és de tipus `application/x-www-form-urlencoded`. Per tant, en aquest escenari les dades queden restringides als codis ASCII. Amb el mètode POST el tipus de contingut es pot escollir i `application/x-www-form-urlencoded` és el tipus de contingut per defecte. Aquest tipus de contingut cal que estigui codificat de la manera següent:

- Els caràcters en blanc es substitueixen per +.
- Els caràcters reservats s'evadeixen tal com es descriu en RFC1738, secció 2.2: els caràcters no alfanumèrics es substitueixen per %HH, on els dos dígitos hexadecimals HH representen el codi ASCII d'aquest caràcter. El trencament de línia es representa pel parell CR LF, és a dir, %0D%0A.
- Les parelles de control nom/valor es separen per &, i el nom del valor es separa amb = .

En la figura (4.1) es veu un exemple de formulari que només suporta el mètode POST. Aquest formulari es troba a

<http://napols.uoc.es/arcgis/rest/services/TrailWalker/FeatureServer/0/addFeatures>.

En la secció 4.3.1 es va comentar que l'operació *Add Features* és suportada per la capa de punts amb identificador 0 del servei

<http://napols.uoc.es/arcgis/rest/services/TrailWalker/FeatureServer>

ArcGIS REST Services Directory [Login](#) | [Get Token](#)

[Home](#) > [services](#) > [TrailWalker \(FeatureServer\)](#) > [Posicio](#) > [addFeatures](#) [Help](#) | [API Reference](#)

Add Features: Posicio (ID: 0)

Note: This operation is only supported via POST

Features:

```
[{"geometry":
  {"x":2.894726,"y":41.827171,"spatialReference":
  {"wkid":104910}},"attributes":
  {"CodiEquip":214190070009393,"Equip":ARDUINO_000,"Bateria":100,"Entrada": "2012-10-22
  10:53:12","Tipus":"Arduino"}}]
```

Geodatabase Version Name:

Rollback on Failure: True False

Format:

```
{
  "addResults": [
    {
      "objectId": 12807,
      "success": true
    }
  ]
}
```

Figura 4.1: Servei REST: operació *Add Features* (realitzada mitjançant el mètode POST).

Aquest exemple permet veure la senzillesa d'un servei REST. A més, realitzant una captura amb l'aplicació *Wireshark*, que és un analitzador de protocols de xarxa, es podrà esbrinar quina és l'estructura de totes les trames intercanviades.

4.4.2 JSON

JSON (*JavaScript Object Notation*) és un format per l'intercanvi de dades lleuger. És fàcil de llegir-lo i escriure'l pels humans i d'analitzar-lo per les màquines. JSON és un format de text independent però que resulta familiar als programadors que treballen amb llenguatges de la família C: C, C++, C#, Java, JavaScript, Perl, Python,... A continuació es mostra l'objecte JSON utilitzat en la secció anterior:

```
[
  {
    "geometry":
    {
      "x":2.894726,
      "y":41.827171,
      "spatialReference":
      {
        "wkid":104910
      }
    }
  }
]
```



```

Accept: text/html, application/xhtml+xml, */*
Referer: http://napols.uoc.es/arcgis/rest/services/TrailWalker/FeatureServer/0/addFeatures
Accept-Language: es
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; Trident/5.0)
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: napols.uoc.es
Content-Length: 374
Connection: Keep-Alive
Cache-Control: no-cache

```

Dades HTTP

```

features=%5B%7B%22geometry%22%3A%7B%22x%22%3A2.894726%2C%22y%22%3A41.827171%2C%22
spatialReference%22%3A%7B%22wkid%22%3A104910%7D%7D%2C%22attributes%22%3A%7B%22Cod
iEquip%22%3A214190070009393%2C%22Equip%22%3AARDUINO_000%2C%22Bateria%22%3A100%2C%
22Entrada%22%3A+%222012-10-22+10%3A53%3A12%22%2C%22Tipus%22%3A%22Arduino%22%7D%7D
%5D&gdbVersion=&rollbackOnFailure=true&f=html

```

4.5 L'API d'ArcGIS per JavaScript

L'API de JavaScript és una manera lleugera d'incrustar mapes i tasques en aplicacions *Web*. Els mapes es poden aconseguir d'ArcGIS Online, el propi servidor ArcGIS o altres servidors. L'API de JavaScript té l'avantatge que no cal instal·lar cap *plugin* al navegador i que pot ser compatible amb els dispositius portàtils.

Es recomana accedir directament a l'API emmagatzemat pel servidor d'ArcGIS de la següent manera:

```

<script type="text/javascript"
src="http://serverapi.arcgisonline.com/jsapi/arcgis/?v=3.1"></script>

```

La raó és que existeix una col·lecció de servidors *Web* repartits geogràficament de manera que distribueixen el contingut de manera eficient. A més, aquests servidors balancegen la càrrega. Per altra banda no és necessari reinstal·lar l'API quan apareixen noves versions.

4.5.1 Dojo

Dojo és una eina de codi obert que ajuda a escriure codi JavaScript robust i eficient. JavaScript és un llenguatge que s'executa dins un navegador *Web*. Els diferents navegadors *Web* interpreten el JavaScript de manera lleugerament diferent. Per tal de no necessitar gestionar aquestes particularitats, es fan servir caixes d'eines tals com Dojo, jQuery, Yui, Prototype, etc, que abstrueixen aquestes diferències i les fan invisibles al programador. Dojo permet programar d'una manera més senzilla i eficient, es redueixen les línies de codi, i el programa és més estable.

L'API d'ArcGIS per JavaScript està construït utilitzant Dojo. Entre les miniaplicacions Dojo (*Dojo widgets* o *dijits*) es troba la barra lliscant del nivell de *zoom* proporcionada pel mateix Dojo o l'*info window* creada pels desenvolupadors d'Esri per a l'API de JavaScript.

Les mínimes funcions de Dojo que es necessiten són:

- `dojo.require`: és similar a una etiqueta `<script>` en una pàgina HTML. Importa un recurs. Per exemple, per importar el recurs `esri.map`:

```
dojo.require("esri.map");
```

A continuació es llisten els recursos més comuns així com els seus usos:

- `esri.map`: mapa, geometria, gràfics i símbols.
 - `esri.layers.agsdynamic`: `ArcGISDynamicMapServiceLayer`.
 - `esri.layers.agstiled`: `ArcGISTiledMapServiceLayer`.
 - `esri.tasks.find`: trobar tasques.
 - `esri.tasks.geometry`: tasques de geometria.
 - `esri.tasks.gp`: tasques de geoprocessament.
 - `esri.tasks.identify`: identificació de tasques.
 - `esri.tasks.locator`: localitzador de tasques.
 - `esri.tasks.query`: recerca de tasques.
 - `esri.toolbars.draw`: dibuixar.
 - `esri.toolbars.navigation`: navegar.
- `dojo.ready` o `dojo.addOnLoad`: Indica la funció d'inicialització que s'executarà després que la pàgina s'hagi acabat de carregar. Per exemple:

```
dojo.ready(init);
```

on `init` és la funció d'inicialització.

- `dojo.connect`: Indica que s'està a l'escolta d'esdeveniments específics sobre un objecte o element de la pàgina i quina és la funció que retornarà el resultat en donar-se aquest esdeveniment. Per exemple:

```
dojo.connect(elMeuMapa, "onLoad", elMeuGestorDeCarrega);
```

En aquest cas, la funció és `elMeuGestorDeCarrega`, l'objecte és `elMeuMapa` i l'esdeveniment és `onLoad`.

- `dojo.byId`: La funció retorna el primer element HTML amb l'identificador indicat en l'argument. Per exemple:

```
dojo.byId("daterange").innerHTML = " Des de: " + ...;
```

En escriure aplicacions en JavaScript és molt avantatjós aprofitar la caixa d'eines de Dojo completa. Inclou botons, graelles, diagrames, diagrames en arbre i molts altres *widgets*. La caixa d'eines està dividida en tres parts.

- Core: les funcions essencials que s'acaben de llistar.
- Dijit: arbres, menús i botons.
- DojoX: gràfics, graelles i diagrames.

4.5.2 Dojo dijits

Els Dojo *widgets* o *dijits* serveixen per afegir components d'interfície per als usuaris: caixes de text, botons, menús, calendaris, etc. El seu comportament és consistent en diferents navegadors. Un mateix pot crear els seus propis *dijits*. Es poden tancar diferents elements en un *dijit* per la seva posterior reutilització.

En el cas de tenir *dijits* que apareixen quan l'aplicació es carrega cal incloure:

```
<script type="text/javascript"> dojoConfig = { parseOnLoad: true }</script>
```

abans d'afegir l'etiqueta `<script>` que indica on trobar l'API de JavaScript per ArcGIS.

4.5.3 Dojo layout dijits

Aquests *dijits* proporcionen una composició flexible dels diferents elements UI (*User Interface*) d'HTML amb un mínim d'esforç. S'utilitza `BorderContainer` de Dojo per posicionar els diferents elements d'una manera senzilla i estàndard.

Cada instància de `BorderContainer` permet 5 regions diferents: *top*, *bottom*, *right*, *left* i *center*.

Cada regió conté un dels següent elements Dojo:

- `AccordionContainer`
- `ContentPane`
- `SplitContainer`
- `StackContainer`
- `TabContainer`

Per trencar una regió en seccions addicionals es poden crear `BorderContainers` dins de `BorderContainers`.

L'atribut de disseny del `BorderContainer` defineix com estan disposats els seus elements. Els valors vàlids per l'atribut de disseny poden ser: *headline* o *sidebar*. En el primer cas les regions *top* i *bottom* tenen prioritats i ocupen el 100% de l'amplada. En canvi, en el segon cas, les regions *left* i *right* s'expandeixen al 100%.

Per les regions *top* i *bottom* només s'ha d'indicar la seva alçada, i per les regions *left* i *right* només s'ha d'indicar la seva amplada. Per la regió central no s'ha d'indicar ni amplada ni alçada ja que `BorderContainer` omplirà l'espai romanent amb aquest element.

Cal tenir en compte les següents consideracions:

- L'atribut *gutters* de `BorderContainer` per defecte és `true` i afegeix un marge de 5px a tots els seus elements.
- Cal utilitzar CSS per assegurar-se que les etiquetes `html` i `body` ocupen la finestra completa.

```
html, body {
    height: 100%;
    width: 100%;
    margin: 0;
    padding: 0;
}
```

- El tema Dojo i color de fons de l'aplicació s'incorpora en l'etiqueta body:

```
<body class="soria"
style="background-color: steelblue; overflow: hidden;">
```

- El primer element dins l'etiqueta body hauria de ser un BorderContainer. S'utilitza l'atribut d'estil per afegir un marge si es desitja.

```
<div dojotype="dijit.layout.BorderContainer" design="headline"
gutters="false" style="width: 96%; height: 95%;margin: 2%;
border:solid 3px silver;">
```

- S'utilitza un ContentPane per la divisió corresponent al mapa.

```
<div id="map" dojotype="dijit.layout.ContentPane" region="center"
style="border:solid thin silver;overflow:hidden;"></div>
```

4.5.4 Dades amb coneixement del temps

ArcGIS 10 suporta capes que tenen coneixement del temps i que emmagatzemen informació que varia amb el temps. Quan aquestes capes existeixen en un mapa, es tracta d'un mapa amb coneixement del temps. Cal establir l'extensió (*extent*) de la dimensió temporal en el mapa.

La manera més senzilla de treballar amb aquest tipus de capes és utilitzar el *widget* TimeSlider que gestiona el procés d'actualitzar la finestra temporal.

Per tal que una capa estigui assabentada del temps, cal anar a *Layer Properties* en l'aplicació ArcMap, clicant amb el botó dret del ratolí sobre la capa. Cal que el camp sigui del tipus *date*.

Per establir l'extensió de temps es pot fer de manera automàtica com per exemple:

```
var layerTimeExtent = results[2].layer.timeInfo.timeExtent;
```

o bé, manualment com es realitza en el següent fragment:

```
var timeExtent = new esri.TimeExtent();
timeExtent.startTime = new Date(2012,09,22,10,0,0,0);
timeExtent.endTime = new Date(2012,09,23,18,0,0,0);
```

on s'ha utilitzat el constructor Date de JavaScript:

```
new Date(year, month, day, hours, minutes, seconds, milliseconds);
```

L'API de JavaScript disposa del *widget* TimeSlider que simplifica el procés de visualitzar dades temporals. Permet visionar dades fins a un cert punt en el temps o en un interval de temps.

Primer cal associar el *time slider* al mapa, que evidentment ha de disposar d'una capa amb coneixement del temps. Després cal definir el nombre de parades que la barra lliscant mostra. En el següent fragment de codi es veu una manera possible d'implementar el *time slider*.

```
var timeSlider = new esri.dijit.TimeSlider({style: "width: 1000px;"},
dojo.byId("timeSliderDiv"));
map.setTimeSlider(timeSlider);
```

```
timeSlider.setThumbCount(2); //2 marcadors
timeSlider.createTimeStopsByTimeInterval(timeExtent,10,'esriTimeUnitsMinutes');
timeSlider.setThumbIndexes([0,1]);
timeSlider.setThumbMovingRate(2000);
timeSlider.startup();
```

Els marcadors indiquen la posició en la barra lliscant. En aquest cas hi ha dos marcadors i es filtraran els punts que estan en l'interval de temps donat pels dos marcadors. L'interval de temps mínim és de 10 minuts. El primer marcador està en la primera posició possible (índex 0) i el segon marcador està posicionat en la següent posició (índex 1), és a dir, al cap de 10 minuts. El *time slider* disposa d'un botó de reproducció. S'ha programat que l'interval de temps fixat pels dos marcadors avanci una posició cada 2 segons.

4.6 El visor Web

En la figura (4.3) es veu l'aplicació *Web* de seguiment dels equips. Es poden observar les següents diferents capes:

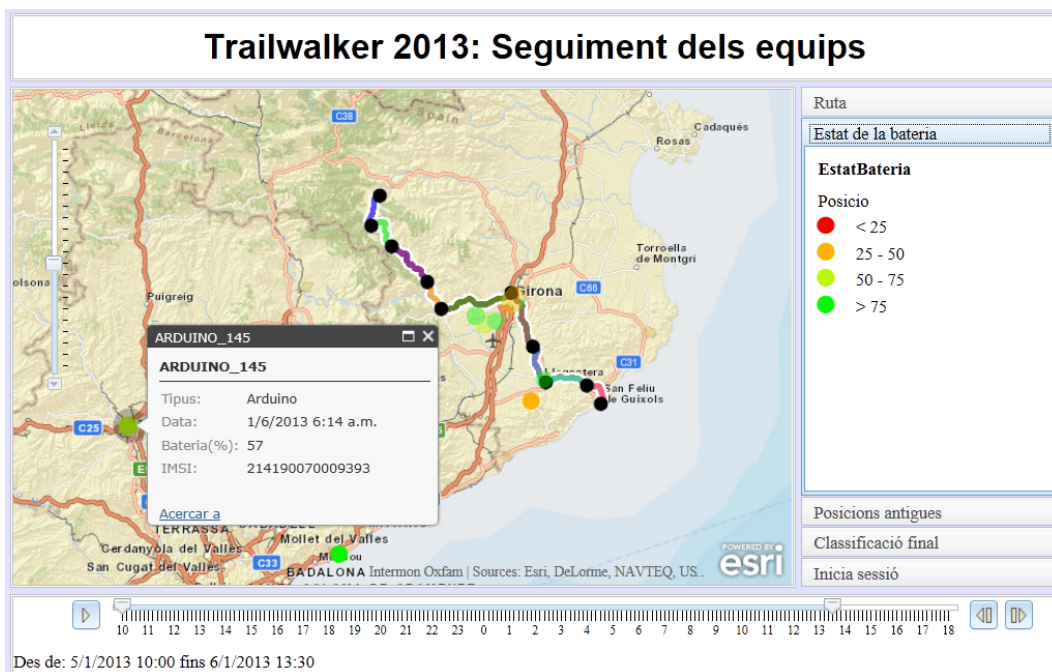


Figura 4.3: Aplicació *Web* de seguiment dels equips

- La capa corresponent al mapa base és de tipus mosaic. La informació corresponent a aquesta capa està emmagatzemada en un servidor d'ArcGIS: http://server.arcgisonline.com/ArcGIS/rest/services/World_Street_Map/MapServer
- En la capa dinàmica de la ruta es poden veure els punts de control de la cursa TrailWalker així com les diferents etapes. En la regió dreta del visor, hi ha una pestanya en la qual es pot veure la llegenda corresponent a aquesta capa. La informació corresponent a aquesta capa s'obté a partir del servei de mapa: <http://napols.uoc.es/arcgis/rest/services/rutes/MapServer>

- També s'utilitza el servei de mapa:

<http://napols.uoc.es/arcgis/rest/services/UltimaEntrada/MapServer>

La capa <http://napols.uoc.es/arcgis/rest/services/UltimaEntrada/MapServer/0> conté la última posició rebuda de cada equip participant.

- Per la capa dinàmica dels punts de seguiment es necessita el servei de mapa:

<http://napols.uoc.es/arcgis/rest/services/TrailWalker/MapServer>

La capa dinàmica dels punts de seguiment s'actualitza cada vegada que un dispositiu mòbil envia una posició al servei *Web FeatureService*:

<http://napols.uoc.es/arcgis/rest/services/TrailWalker/FeatureServer/0>

Aquesta aplicació també disposa d'altres funcionalitats:

- En la llegenda de la regió dreta del visor *Web* es pot observar que el color del símbol dels punts indica l'estat de la bateria depenent del percentatge de bateria restant. L'aplicació també disposa d'una llegenda corresponent a la capa que inclou els diferents trams de la ruta i els punt de control entre els trams.
- Inicialment només es disposava de la capa amb tots els punts de tots els equips. La primera intenció era utilitzar un *Time Slider* per donar un sentit a aquesta gran quantitat de punts provinents de molts equips i de molts instants diferents. Amb aquesta idea es va construir una primera versió del visor *Web*. Després es va afegir un nou servei de mapa amb una capa on només hi ha l'última entrada de cada equip. En l'actual versió de l'aplicació, en el panell de la dreta, en la pestanya *Posicions antigues* es disposa d'una barra lliscant, que permet tornar transparents totes les posicions excepte l'última posició enviada per cada equip.
- En el panell inferior es veu el *time slider* i també s'indica quina és exactament la finestra temporal que s'està visualitzant en el mapa.
- En clicar sobre un punt, s'obre una finestra amb informació: nom de l'equip, codi de l'equip (IMSI), l'hora i la data en què l'equip va recollir la posició, percentatge de bateria restant, i tipus de dispositiu (Arduino, Android, etc).

Capítol 5

Línies futures de desenvolupament

Aquest és un treball de final de carrera força extens. Havent disposat de més temps es podrien haver desenvolupat amb detall les següents línies futures de desenvolupament:

5.1 Dispositiu mòbil

- És essencial reduir el consum. És interessant poder adormir el microcontrolador i el mòdul SIM5218 de la placa 3G/GPRS tal com s'ha indicat en 2.6.2 i 2.6.2 respectivament.
- Directament les piles no es poden posar en paral·lel. Però es poden investigar les opcions que existeixen per posar bateries en paral·lel de manera eficient. Així es podrien assolir capacitats de corrent majors. El desavantatge seria el pes, ja que de 5 o 6 piles AA es passaria a 10 o 12 piles, si per exemple es volgués duplicar la capacitat de corrent.
- Cal realitzar més proves al dispositiu mòbil. En el cas de tenir una alimentació indefinida, s'ha d'assegurar que el funcionament del dispositiu es superior a 32 hores. Cal fer proves per comprovar que sempre és així i depurar les possibles causes d'error. S'han depurat unes quantes causes d'errors, i finalment s'han assolit les 32 hores. Però són necessàries més proves per descartar que es produeixin més errors d'aquest tipus. Els errors que s'han detectat i solucionat, han estat:
 - S'ha observat un curiós funcionament del GPS en ocasions. Per exemple, en algun cas, després de funcionar, a intervals de 10 minuts, unes 4 hores correctament, es va observar que va deixar d'adquirir noves dades i va començar a repetir sempre les dades corresponents a l'últim posicionament correcte. S'ha intentat realitzar un control suficient d'errors, per evitar tots aquests casos. En un primer moment es va programar el codi de manera que és reinicialitzés en detectar el problema. Finalment, veient que després de la reinicialització el GPS no podia posicionar-se, ni proporcionar cap dada, s'ha deduït que aquest problema es dona per realitzar les proves des de l'interior i s'ha eliminat el codi que efectuava aquest control. Probablement, aquest problema no es donarà a l'exterior.
 - S'ha optat per una reinicialització del mòdul en aquells punts de codi on s'ha observat en varies ocasions bucles infinits. Aquesta reinicialització es realitzarà si es detecta aquest tipus de bucle. De totes maneres, després de moltes depuracions, aquest tipus d'error ja no s'ha donat més.

- Altres vegades, la placa Arduino ha arrancat de nou per raons desconegudes, estant en un punt on el codi que s’havia d’executar era molt senzill i no podia provocar problemes.
 - * La comanda AT+CPOF ha estat de gran ajuda per tal que el dispositiu arranqués correctament. Si no s’apagués el mòdul SIM5218 i després es tornés a engegar, comandes com AT+CPIN i AT+CGSOCKCONT causarien problemes. Per exemple, el programa es quedaria bloquejat en espera d’OK com a resposta de la comanda AT+CPIN. La raó és que el mòdul ja té el PIN introduït. En tornar-lo a introduir, el mòdul entén que s’està introduint el PIN2. Però el codi introduït és incorrecte ja que es tracta del codi del PIN1. Per tant la resposta a la comanda no conté la paraula OK sinó la paraula ERROR.
 - * Finalment s’ha descobert que la causa que provocava la reinicialització de la placa Arduino era que la memòria SRAM era insuficient.
- Seria interessant realitzar proves a l’exterior de llarga durada. Les úniques proves que s’han portat a terme són amb un senzill adaptador per piles del tipus HR22. En aquest cas s’ha utilitzat un pila recarregable de 170 mA · h. Cal realitzar un adaptador que permeti utilitzar piles de tipus AA.

5.2 Servei Web

El servidor s’hauria de programar per a què realitzés les següents accions:

- Calcular la distància d’un punt d’un equip respecte la meta, seguint la ruta. L’algoritme no seria gaire complicat. Bàsicament, primerament s’hauria de calcular quin és el punt de la ruta més proper al punt de l’equip, i després trobar la distància d’aquest punt de la ruta a la meta. A més, les distàncies dels punts de la ruta respecte el punt de la meta ja poden estar prèviament calculades. Es podria refinar el procediment realitzant alguna interpolació de punts perquè es tingués una precisió més acurada.
- Estimar la posició en què es troba cada equip, a partir de les diferents posicions rebudes de l’equip. Si un equip envia la seva posició cada 10 minuts, la posició estimada serà una mica més endavant en la ruta. Es poden considerar diferents posicions anteriors per estimar la velocitat de l’equip i la posició actual amb la màxima precisió possible.
- Estimar la distància, respecte la meta i seguint la ruta, en què es troba cada equip, a partir de la posició estimada de l’equip. Aquesta informació serà essencial per poder realitzar una estimació de les classificacions en qualsevol instant de la cursa. En els punts de control al llarg del recorregut es podria prescindir d’anotar els temps d’arribada dels diferents equips. En el punt de control final a la meta, serà necessari igualment anotar els diferents temps d’arribada. Les estimacions realitzades no tindran evidentment una precisió suficient per la classificació final.
- Estimar altra informació que pot ser d’interès com: la posició que ocupa un equip en la classificació, distància recorreguda i temps necessari per arribar a la meta.
- Caldria afegir un servei al directori de serveis per tal que es poguessin afegir punts amb el nom de l’equip i l’hora d’arribada a la meta. Evidentment les coordenades serien les de la meta.

- El serveis *Web* han de ser segurs. Tot i que és poc probable, qualsevol persona pot enviar punts des de casa seva. Es necessari un sistema d'autenticació segur dels equips.

5.3 Visor Web

Es podrien afegir moltes funcionalitats:

- Seria interessant afegir un buscador d'equips a partir del nom de l'equip. En el moment que es cliqués sobre un equip es podrien veure tots els punts de l'equip o l'últim punt o bé es podria utilitzar el *Time Slider*. També es podria permetre la selecció de múltiples equips per la visualització de les posicions.
- A partir del que s'ha proposat en 5.2 es podrien llistar les classificacions provisionals en qualsevol instant de temps, actual o passat.
- Per la qualificació final, s'hauria de crear la possibilitat d'autenticació de les persones que realitzen el control en la meta. Aquestes haurien de poder, de manera senzilla, afegir dades en una capa d'un servei del directori de serveis que disposi de l'operació *Add Features* i *Delete Features*. S'hauria de preparar l'aplicació *Web* per a què només s'hagués d'especificar el nom de l'equip. L'hora es podria posar de manera automàtica o bé de manera manual. S'hauria de proporcionar també la manera de rectificar una introducció d'un nom d'equip equivocat.
- Donat un punt d'un equip sobre el mapa, en clicar-lo, es pot veure la informació corresponent al punt. Es podria millorar de manera que es pogués deixar un missatge donant ànims a l'equip corresponent al punt. També es podria demanar informació de la posició estimada de l'equip, de la distància que li queda fins a la meta i del temps que probablement trigarà en arribar a la meta.

Capítol 6

Conclusions

6.1 Objectius assolits

Els objectius a assolir en aquest treball de final de carrera han estat proporcionar:

1. Un dispositiu de seguiment que utilitzi GPS i GPRS i que sigui apte per un equip participant en una cursa de 32 hores de durada (Trailwalker).
2. Un servidor per recollir les dades enviades per qualsevol dispositiu mòbil de qualsevol equip participant.
3. Un visor *web* per aquestes dades que mostri les posicions dels equips tant per als seguidors de la cursa com per als mateixos participants i que, a més, mostri la classificació final.

Tenint en compte l'extensió que pretén abastar es pot considerar com un treball ambiciós. Apropar-se a un assoliment complet dels objectius ha estat complicat. L'èxit obtingut en els tres punts anteriors ha estat gairebé complet.

1. El dispositiu funciona correctament, però el seu consum és bastant elevat. S'ha intentat pensar en una bateria econòmica. L'opció més senzilla i de menor consum és apagar el mòdul 3G/GPRS i enviar dades cada 10 minuts. És el programari qui apaga el mòdul. Evidentment, els participants de la cursa no han de fer res.

En un primer estudi es va veure que amb 5 o 6 piles AA alcalines de 2900 mA · h en sèrie, es podria arribar a 27 hores i 35 minuts. Si les dades s'envien cada 20 minuts es podria arribar a 30 hores i 51 minuts. Tenint en compte que la majoria dels equips acaben la cursa en 24 hores es pot considerar com una solució viable.

En un estudi posterior es va veure que els 7.5 V que proporcionen 5 piles o els 9 V que proporcionen 6 piles, no són suficients. Els pics elevats d'intensitat que provoca la placa 3G/GPRS, condueixen a què la placa 3G/GPRS deixi de funcionar ben aviat. En augmentar la resistència interna de la bateria degut al desgast, durant un pic d'intensitat no queda la tensió mínima perquè la placa 3G/GPRS funcioni correctament. Per tant s'hauria d'augmentar el voltatge total al màxim recomanat de 12 V.

La conclusió final és que s'haurien d'utilitzar 8 piles alcalines AA en sèrie de 2900 mA · h. L'aplicació hauria d'apagar el GPS, quan no es fes servir, i també apagar el mòdul SIM5218 quan es realitzés una espera. S'ha comprovat que amb una espera de 10 minuts aquesta bateria es suficient per tota la cursa.

2. El servei *Web* funciona perfectament. S'ha de dir que aquesta part l'ha realitzada en Francesc Puigvert per tres raons:
 - Aquest treball de final de carrera ha estat molt extens.
 - Els camps havien ser els mateixos per tots els estudiants realitzant el programari per algun dispositiu.
 - El dispositiu Arduino va arribar amb un endarreriment considerable.
3. El visor mostra la posició dels diferents equips, a més inclou la capa de ruta, una llegenda i un *Time Slider*. No mostra la classificació final. La realitat és que el tercer objectiu tenia un problema en el plantejament. Per poder realitzar la classificació final, els dispositius mòbils han d'enviar cada vegada les dades més ràpid a mida que s'apropen a la meta i en el moment d'arribar a la meta hen de deixar d'enviar de punts. Això té dues implicacions.
 - El punt de control de la meta ha d'estar decidit prèviament ja que tots els dispositius mòbils han de portar incorporades les seves coordenades en el programari. A més no es pot canviar a última hora la posició de la meta. En el programari del dispositiu mòbil Arduino s'ha incorporat l'opció de detectar la meta i enviar les dades amb una freqüència superior a mesura que l'equip s'acosta a la meta. De totes maneres es pot configurar fàcilment perquè no detecti la meta i enviï les dades sempre cada 10 minuts o cada qualsevol altre valor de temps que es desitgi.
 - El programari dels diferents dispositius ha de portar incorporada una funcionalitat similar.

Finalment es va optar perquè fossin els encarregats del punt de control de la meta qui entressin l'hora d'arribada a la meta de cada equip. Davant d'aquest canvi i l'ampliació en la quantitat de programació del visor *Web* que suposava, aquest punt va ser posposat.

6.2 Depuració

La depuració és costosa i a vegades frustrant. Es percep en forma d'aleatorietat. Una petita modificació en el codi pot produir que aquest no funcioni. Un aparent `delay()` innecessari era absolutament necessari. Aconseguir un funcionament seguit de 32 hores utilitzant un adaptador AC i el port USB d'un ordinador, ha estat difícil. A vegades el GPS fallava sense cap raó, pot ser perquè estava en l'interior d'un edifici al costat d'una finestra i la configuració de la constel·lació de satèl·lits en aquell moment era molt poc favorable, pot ser per una altra raó. A vegades era l'Arduino qui es reinicialitzava sense cap motiu aparent, ja que les dades s'havien transmès correctament al servidor i, aparentment, només tenia per endavant 10 minuts de calma completa. Les causes d'aquesta dificultat gran en la depuració són diverses.

6.2.1 Disseny del dispositiu mòbil

El dispositiu mòbil té una deficiència de disseny. Les conseqüències d'aquest disseny són una inversió de temps molt més gran en tasques de depuració.

La placa Arduino té només un pin RX i un pin TX. Aquests pins s'utilitzen per la comunicació amb la placa 3G/GPRS, però també s'utilitzen a la vegada per les tasques de depuració. Cada vegada que l'Arduino executa una línia de codi

```
Serial.print("missatge");
```

la informació viatja de la placa Arduino al port USB i, també, de la placa Arduino al mòdul 3G/GPRS. Per tant, tots els missatges de depuració van al mòdul 3G/GPRS. Això pot ser causa de molts problemes: targetes SIM amb el PIN2 bloquejat, *sketchs* que no funcionen perquè els missatges de depuració estaven en un lloc no gaire afortunat, etc. Al final tot es va aprenent però a costa de consumir molt temps:

- Es minimitzen els missatges de depuració.
- S'evita que apareguin caràcters que poden formar part d'una comanda AT, com el caràcter +. Per si de cas, s'eviten altres caràcters de control, dels quals es desconeixen les conseqüències del seu ús.
- S'intenta trobar llocs dins el codi que aparentment no semblin crítics. Per exemple, quan ja s'ha parat el GPS es poden visualitzar les dades que ha capturat, o quan ja s'han transmès les dades al servidor es pot visualitzar el contingut d'un *buffer* que emmagatzema informació i et permet veure per exemple l'identificador de l'objecte creat pel servidor.

En el cas d'haver dissenyat la placa 3G/GPRS per l'Arduino Mega en lloc de l'Arduino UNO, aquest problema no hauria existit, ja que disposa de més pins RX i TX. En aquest cas, les dimensions del dispositiu serien majors. L'Arduino Mega mesura 101.6mm x 53.3mm, mentre que l'Arduino UNO 68.6mm x 53.3mm, és a dir, uns 3 cm menys de llargada.

6.2.2 El mòdul SIM5218

Les dificultats en la depuració s'incrementen quan afegim que el mòdul SIM5218 necessita un marge de temps pel bon funcionament, que s'ha de respectar. El problema és que aquest marge de temps no està clarament definit ni documentat. Si que està documentat que abans d'enviar una segona comanda AT al mòdul SIM5218, cal esperar haver rebut la resposta de la primera comanda. Però a vegades necessita un marge de temps més gran que cal descobrir-lo provant. Per exemple, quan s'està intentant esbrinar les coordenades cal fer unes quantes peticions AT+CGPSINFO per tal de poder-se posicionar. Si aquestes peticions es realitzen només esperant la resposta a la comanda i sense un marge entre peticions d'un mínim d'aproximadament un segon es produeix un bloqueig en executar la següent comanda, que és AT+CHTTPACT. S'espera una resposta que no arriba i es produeix un bucle infinit. En canvi si s'espera 1 o 1.5 segons després de rebre la resposta d'una petició AT+CGPSINFO i abans de tornar a realitzar una altra petició AT+CGPSINFO, aquest bloqueig no es produeix.

6.2.3 Mida de la SRAM

La mida de la SRAM és 2 KB. En el cas, que no hi hagi cobertura 3G/GPRS, les dades s'emmagatzemen. Esbrinar quina és la capacitat màxima no és fàcil. He optat per emmagatzemar 10 dades. Aquest és un valor força ajustat. A partir de proves es descobreixen els símptomes d'una SRAM plena:

- L'execució del codi queda aturada en un punt. Aquest punt pot ser, per exemple, just abans de la crida a una funció. No hi ha prou SRAM per poder efectuar la crida.

- L'Arduino reinicialitza l'execució del codi i torna a executar la funció `setup()`. Aquesta reinicialització la decideix la placa Arduino, no el codi instal·lat.

Una senzilla decisió d'eliminar codi repetit i crear una funció pot tenir com a conseqüències, un funcionament no desitjat del codi. Aleshores, és necessari disminuir aquesta sobrecàrrega de la SRAM per una altra banda.

6.3 Valoració de la planificació

L'endarreriment aproximat d'un mes i mig que ha patit el lliurament del dispositiu mòbil, ha dificultat en una gran mesura qualsevol possible seguiment d'una planificació. Ha estat necessari canviar el punt de vista. L'únic principi que s'ha utilitzat és el d'intentar avançar en algun sentit.

- S'ha intentat aprofitar totes les estones possibles, per petites que fossin.
- S'ha tingut la previsió de tenir sempre material a punt per poder realitzar qualsevol tipus de tasca necessària.

L'aplicació del dispositiu mòbil està escrita gairebé al 100% utilitzant llenguatge C. Això ha estat d'una gran ajuda, ja que totes les funcions que no interactuen amb la placa 3G/GPRS han estat prèviament provades en un entorn de programació per C (eclipse), sense necessitat de disposar del dispositiu mòbil. El contratemps d'aquest endarreriment ha estat favorable en altres aspectes:

- És més fàcil depurar en l'entorn de programació eclipse.
- Com a conseqüència d'intentar depurar la màxima quantitat de codi possible, el codi ha quedat més ben estructurat en funcions.
- S'ha pogut depurar sense les esperes que suposen l'adquisició de dades del GPS.
- Tampoc ha estat necessari estar just al costat d'una finestra ni esperar que la posició dels satèl·lits fos adequada per rebre les coordenades. Hi ha intervals de temps a vegades d'hores, en què el GPS no pot proporcionar les coordenades tot i estar al costat d'una finestra.

Altres aspectes que han estat causa de molts endarreriments.

- L'amplitud desorientadora d'ArcGIS.
- La dificultat de depurar el codi instal·lat en el dispositiu mòbil i esbrinar quina és la causa entre les múltiples possibles causes d'error.

De totes maneres, els dos lliuraments parcials s'han efectuat en els terminis previstos:

- El primer lliurament contenia tot el que estava relacionat amb el dispositiu mòbil: codi de l'aplicació i documentació. Evidentment el codi no s'havia pogut provar en el propi dispositiu, però un 80% del codi funcionava correctament.
- El segon lliurament contenia tot el que estava relacionat amb ArcGIS: codi de l'aplicació del visor *Web* i documentació.

La depuració de l'aplicació del dispositiu mòbil ha estat tan costosa que s'ha allargat fins a la data de lliurament de la memòria, conjuntament amb l'estudi del tipus de bateria necessària. També s'han aprofitat els darrers dies per millorar el visor *Web*.

Apèndix A

Appèndix

A.1 Mesures a la terra

A.1.1 Radi de la terra

La terra no és una esfera, és un el·lipsoide. El radi¹ equatorial és de $a = 6378.137$ km i el radi polar és de $b = 6356.7523142$ km. Per tal de saber si el dispositiu mòbil sobre el qual s'està realitzant un seguiment, ha arribat o no a la meta, és necessari calcular el radi de la terra en la latitud corresponent al punt de meta. Sigui φ la latitud en aquest punt. Es realitza una secció vertical passant pel punt i s'obté una el·lipse. Aquesta el·lipse es pot definir per

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$$

Escrivint $(x, y) = (r \cos \varphi, r \sin \varphi)$, s'obté que:

$$\left(\frac{r \cos \varphi}{a}\right)^2 + \left(\frac{r \sin \varphi}{b}\right)^2 = 1$$

D'aquí s'aïlla fàcilment

$$\begin{aligned} r &= \sqrt{\left(\frac{\cos^2 \varphi}{a^2} + \frac{\sin^2 \varphi}{b^2}\right)^{-1}} = \sqrt{\left(\frac{\cos^2 \varphi}{a^2} + \frac{1 - \cos^2 \varphi}{b^2}\right)^{-1}} \\ &= \sqrt{\frac{1}{\frac{1}{b^2} - \left(\frac{1}{b^2} - \frac{1}{a^2}\right) \cos^2 \varphi}} = \sqrt{\frac{b^2}{1 - \left(1 - \frac{b^2}{a^2}\right) \cos^2 \varphi}} \end{aligned}$$

Per tant obtenim que

$$r = \frac{b}{\sqrt{1 - e^2 \cos^2 \varphi}}$$

on

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

s'anomena excentricitat.

¹<http://en.wikipedia.org/wiki/Latitude>

A.1.2 Distància entre dos punts de la terra

En la figura (A.1)² es pot veure un triangle esfèric. En aquest cas la fórmula a utilitzar de trigonometria esfèrica serà:

$$\cos c = \cos a \cos b + \sin a \sin b \cos \gamma \quad (\text{A.1})$$

on a és l'angle que formen els vectors \vec{OB} i \vec{OC} , b és l'angle que formen els vectors \vec{OC} i \vec{OA} , c és l'angle que formen els vectors \vec{OA} i \vec{OB} . O és el centre de la terra. Siguin A i B els dos punts de la superfície esfèrica dels quals es vol calcular la distància. El punt C és el pol nord. Siguin (φ_A, ϕ_A) i (φ_B, ϕ_B) , les coordenades latitud i longitud en radians dels punts A i B respectivament. Aleshores de (A.1) es dedueix que

$$\cos c = \cos \left(\frac{\pi}{2} - \varphi_A \right) \cos \left(\frac{\pi}{2} - \varphi_B \right) + \sin \left(\frac{\pi}{2} - \varphi_A \right) \sin \left(\frac{\pi}{2} - \varphi_B \right) \cos (\phi_A - \phi_B)$$

És a dir

$$\cos c = \sin \varphi_A \sin \varphi_B + \cos \varphi_A \cos \varphi_B \cos (\phi_A - \phi_B)$$

Pert tant

$$c = \arccos (\sin \varphi_A \sin \varphi_B + \cos \varphi_A \cos \varphi_B \cos (\phi_A - \phi_B))$$

Finalment la distància d entre A i B és

$$d = R \arccos (\sin \varphi_A \sin \varphi_B + \cos \varphi_A \cos \varphi_B \cos (\phi_A - \phi_B))$$

En el cas que les unitats de les coordenades (φ_A, ϕ_A) i (φ_B, ϕ_B) , siguin graus, la fórmula per la distància d serà

$$d = R \arccos \left(\sin \left(\frac{\pi \varphi_A}{180} \right) \sin \left(\frac{\pi \varphi_B}{180} \right) + \cos \left(\frac{\pi \varphi_A}{180} \right) \cos \left(\frac{\pi \varphi_B}{180} \right) \cos \left(\frac{\pi (\phi_A - \phi_B)}{180} \right) \right)$$

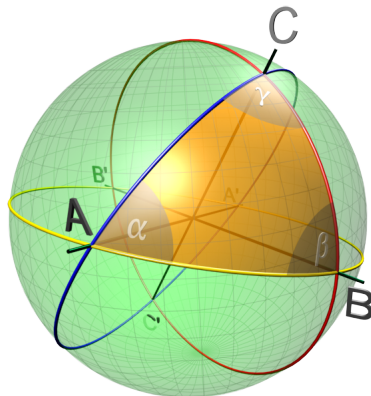


Figura A.1: Triangle esfèric.

²http://commons.wikimedia.org/wiki/File:Spherical_triangle_3d.png

Bibliografia

- [1] *Wikipedia, la enciclopedia libre* [en línia]: *Sistema de posicionamento global*. <<http://es.wikipedia.org/wiki/Gps>>. 7 nov 2012. [Consulta: 10 nov. 2012]
- [2] *Wikipedia, la enciclopedia libre* [en línia]: *GPS Asistido*. <http://es.wikipedia.org/wiki/GPS_Asistido>. 22 oct 2012. [Consulta: 10 nov. 2012]
- [3] *Tienda espía del detective privado PARADETECTIVES* [en línia]: *localizadores GPS espías para coches, motos, barcos y personas*. <<http://www.paradetectives.es/12-gps>>. [Consulta: 10 nov. 2012]
- [4] *Arduino* [en línia]. <<http://arduino.cc/en/>>. 22 oct 2012. [Consulta: 10 nov. 2012]
- [5] *Arduino* [en línia]: *Arduino Uno*. <<http://arduino.cc/en/Main/ArduinoBoardUno>>. [Consulta: 10 nov. 2012]
- [6] BANZI, Massimo. *Getting started with Arduino*. 1a ed. Sebastopol: Make: Books, 2009. ISBN 978-0-596-15551-3
- [7] *Cooking Hacks* [en línia]: *New 3G + GPS shield for Arduino*. <<http://www.cooking-hacks.com/index.php/documentation/tutorials/arduino-3g-gprs-gsm-gps>>. [Consulta: 10 nov. 2012]
- [8] *Atmel* [en línia]: *Atmel 8-bit Microcontroller with 4/8/16/32 KBytes In-System Programmable Flash*. Atmel Corporation, jul. 2012, pàg. 469 (fig. 30-284) <<http://www.atmel.com/Images/doc8271.pdf>>. [Consulta: 29 des. 2012]
- [9] *Lingua-Systems* [en línia]: *ASCII to Unicode Mapping*. <<http://www.lingua-systems.com/knowledge/unicode-mappings/ascii-to-unicode.html>>. [Consulta: 10 nov. 2012]
- [10] *SIMCOM, a company of SIM Tech* [en línia]: *SIM5218*. <<http://wm.sim.com/producten.aspx?id=1026>>. [Consulta: 10 nov. 2012]
- [11] *SIM5218_SERIAL_AT* [en línia]: *Command Manual_V1.21*. <<http://wm.sim.com/downloaden.aspx?id=2984>>. Shangai: SIMCom Wireless Solutions, 2011. [Consulta: 10 nov. 2012]
- [12] *Wikipedia, the free encyclopedia* [en línia]: *Spherical trigonometry*. <http://en.wikipedia.org/wiki/Spherical_trigonometry>. 4 Nov 2012. [Consulta: 10 nov. 2012]
- [13] *World Wide Web Consortium W3C* [en línia]: *HTML 4 Specification References*. <<http://www.w3.org/TR/html401/interact/forms.html#h-17.13>>. [Consulta: 26 nov. 2012]

- [14] Pilgrim, Mark. *Dive into HTML5* [en línea]: *Dive into HTML5*. <<http://diveintohtml5.info/>>. [Consulta: 7 des. 2012]
- [15] *ArcGIS* [en línea]: *ArcGIS Resources* <<http://resources.arcgis.com/en/home/>>. [Consulta: 7 des. 2012]
- [16] *ArcGIS* [en línea]: *ArcGIS JavaScript API* <<http://help.arcgis.com/en/webapi/javascript/arcgis/index.html>>. [Consulta: 7 des. 2012]
- [17] *ArcGIS* [en línea]: *ArcGIS JavaScript Samples* <http://help.arcgis.com/en/webapi/javascript/arcgis/help/jssamples_start.htm>. [Consulta: 7 des. 2012]
- [18] *ArcGIS* [en línea]: *Code Gallery* <http://help.arcgis.com/en/webapi/javascript/arcgis/help/gallery.html>>. [Consulta: 7 des. 2012]
- [19] *JSON* [en línea]: *Introducing JSON* <<http://www.json.org/>>. [Consulta: 10 nov. 2012]
- [20] *Wikipedia, The Free Encyclopedia* [en línea]: *Web Service* <http://en.wikipedia.org/wiki/Web_service>. 4 des. 2012 [Consulta: 11 des. 2012]
- [21] *Wikipedia, The Free Encyclopedia* [en línea]: *SOAP* <<http://en.wikipedia.org/wiki/SOAP>>. 1 des. 2012 [Consulta: 11 des. 2012]
- [22] *Wikipedia, The Free Encyclopedia* [en línea]: *Representational state transfer* <http://en.wikipedia.org/wiki/Representational_state_transfer>. 23 nov. 2012 [Consulta: 11 des. 2012]
- [23] Amodeo Rubio, Enrique. *WordPress.com* [en línea]: *Servicios Web (2): ¿Qué es REST?* <<http://eamodeorubio.wordpress.com/2010/07/26/servicios-web-2-%C2%BFque-es-rest/>>. 26 jul. 2010 [Consulta: 7 des. 2012]