



Sii-Mobility

Supporto di Interoperabilità Integrato per i Servizi al Cittadino e alla Pubblica Amministrazione

Trasporti e Mobilità Terrestre, SCN_00112

Deliverable ID: DE3.1b

**Titolo: Kit veicolari - specifica di dettaglio dello
sviluppo**

Data corrente	30-06-2017
Versione (solo il responsabile puo' cambiare versione)	v1-15
Stato (draft, final)	finale
Livello di accesso (solo consorzio, pubblico)	Pubblico quando completo
WP	OR3
Natura (report, report e software, report e HW..)	report
Data di consegna attesa	M17 – Maggio 2017
Data di consegna effettiva	
Referente primario, coordinatore del documento	Alessandro Paganone, ECM, alessandro.paganone@ecmre.com
Contributor	Carlo Alberto Dami, CReAI, carloalberto.dami@tin.it Andrea Rindi, UNIFI-DIEF/Mdmlab, andrea.rindi@unifi.it Giovanni Collodi, MIDRA, collodi@ing.unifi.it Leonardo Fabbri, ELFI,

	leonardo.fabbri@elfisrl.net Francesco Chiti, UNIFI-LART, francesco.chiti@unifi.it Laura Cocone, MIZAR, laura.cocone@swarco.com Paolo Nesi, UNIFI, paolo.nesi@unifi.it
Coordinatore responsabile del progetto	Paolo Nesi, UNIFI, paolo.nesi@unifi.it

Sommario

1	Executive Summary	5
2	Terminologia, contesto, riferimenti e notazioni	5
2.1	Acronimi, sigle, terminologia.....	5
2.2	Documenti di riferimento	6
2.2.1	Standards e normative applicabili	6
2.2.2	Documenti Sii-Mobility	6
2.2.3	Altri riferimenti	6
2.3	Contesto.....	7
2.4	Notazioni	8
2.4.1	Schemi di architettura	8
2.4.2	Schemi di dettaglio.....	8
3	Kit bike.....	9
3.1	Architettura generale	9
3.2	Specifica di dettaglio	10
3.2.1	Diagramma di contesto generale.....	10
3.2.2	Piattaforma, KIT01	11
3.2.3	Funzione ATF, anti-theft feature	23
3.2.4	Smartphone utente.....	24
3.2.5	Sensori / Attuatori	25
3.2.6	Protocollo base Bluetooth	37
4	Kit carbus base	39
4.1	Architettura generale	39
4.2	Specifica di dettaglio	40
4.2.1	Diagramma di contesto	40
4.2.2	Interfaccia GUIDO – Sii-Mobility.....	40
4.2.3	Nuovo SW.....	41
4.2.4	Installazione di TURMS	41
5	Kit carbus evoluto	43

5.1	Architettura generale	43
5.2	Specifica di dettaglio	44
5.2.1	Diagramma di contesto generale	44
5.2.2	Piattaforma, KIT03	45
5.2.3	Tablet di bordo	53
5.2.4	Sensori.....	54
5.2.5	Protocollo base Bluetooth	60
6	Appendice A – Studio di fattibilità della funzione AATF (autonomous anti-theft feature) realizzata nel contesto del kit bike	60
7	Bibliografia	61

Figura 1	: i Kit veicolari nel contesto di Sii-Mobily	7
Figura 2	: notazione grafica generale	8
Figura 3	: notazione grafica di dettaglio	8
Figura 4	: Architettura generale del Kit bike	9
Figura 5	: Kit Bike - diagramma di contesto.....	10
Figura 6	: Piattaforma – diagramma di contesto	11
Figura 7	: Kit bike – Contenitore e disposizione apparati.....	13
Figura 8	: Schema generale del kit bike.....	17
Figura 9	: Vehicular Kit Manager Bike – casi d’uso	20
Figura 10	: Vehicular Kit Manager Bike – FSM	21
Figura 11	: Smartphone – diagramma di contesto	24
Figura 12	: modulo GPS VK2828U7G5LF V1.0	26
Figura 13	: Scheda GPS nel contesto della architettura del Kit bike	28
Figura 14	: Principio di funzionamento del sensore ultrasonico con particolare attenzione verso il posizionamento del sensore	32
Figura 15	: modulo sensore di temperatura DS18B20.....	32
Figura 16	: resistenza di pull-up interna al modulo DS18B20.....	33
Figura 17	: Traslatore di livello logico.....	34
Figura 18	: Logica dell’architettura del kit anti collisione.....	35
Figura 19	: Architettura del sistema (KIT anticollisione).....	35
Figura 20	: tre sensori acustici scelti come possibile elemento sensibile ai fini del rilevamento della prossimità tra bike e veicolo che segue.....	36
Figura 21	: Architettura generale del Kit carbus base.....	39
Figura 22	: Kit carbus base - diagramma di contesto.....	40

Figura 23 : Architettura generale del Kit carbus evoluto.....	43
Figura 24 : Kit carbus evoluto - diagramma di contesto.....	44
Figura 25 : Piattaforma – diagramma di contesto	45
Figura 26 : Schema generale del kit carbus evoluto	48
Figura 27 : Carbus Kit Manager Bike – casi d’uso.....	51
Figura 28: Carbus Kit Manager Bike – FSM.....	52
Figura 29 : Smartphone – diagramma di contesto	53
Figura 30 : Connettori OBD-II – tipo “a” (sinistra) e tipo “b” (destra)	55
Figura 31 : UART Adapter per CanBus Sniffer	55

1 Executive Summary

Il documento contiene il progetto dettagliato dei kit veicolari previsti nel progetto:

- Kit bike – realizzato interamente nel contesto del progetto;
- Kit Carbus Base – realizzato utilizzando un’applicazione già esistente e integrandola;
- Kit Carbus Evoluto - realizzato interamente nel contesto del progetto.

Per ciascuno dei kit sono dettagliati:

1. l’architettura generale, contestualizzata in Sii-Mobility
2. l’architettura di dettaglio, con individuazione delle principali componenti, fino al livello di singola scheda elettronica e/o componente software principale
3. la specifica di dettaglio dell’hardware;
4. la specifica di dettaglio del software;
5. la specifica di dettaglio di contenitori e parti meccaniche
6. la sensoristica utilizzata. Per ciascun sensore sono dettagliate le interfacce fisiche e logiche e l’architettura interna. Per alcuni sensori sviluppati nel contesto della Attività 3.2 si rimanda alla relativa documentazione ([SiiM 4] e/o [SiiM 13]).

Questo documento (DE3.1b) ha lo scopo di descrivere quanto si intende sviluppare come prototipo al termine delle fasi di progetto. Altri documenti sono previsti per monitorare lo sviluppo dei prototipi ([SiiM 11]), per fornire la manualistica di configurazione e installazione ([SiiM 10]) e i risultati della sperimentazione ([SiiM 12]). Lievi variazioni in corso d’opera saranno descritte direttamente nei documenti ora menzionati, in caso di variazioni di maggior rilievo sarà possibile una ri-emissione di questo documento.

2 Terminologia, contesto, riferimenti e notazioni

2.1 Acronimi, sigle, terminologia

API	Application Program Interface
AVL - AVM	Automatic Vehicle Location - Automatic Vehicle Monitoring
GPRS	General packet radio service
GPS	Global positioning System
GSM	Global System for Mobile
ICT	Information and Communication Technologies
ITS	Intelligent Transport Systems
LCD	liquid-crystal display
OD	Open Data
RFID	Radio Frequency IDentification o Identificazione a radio frequenza
SII	sistema di interoperabilità integrato
SMS	Short Message Service
SN	social networking, oppure sensor network
SSAMM	Agenzia per la Mobilità Metropolitana strumenti di supporto, TOSCANA
TPL	gestore trasporto pubblico locale

UMTS	Universal Mobile Telecommunications System
VWSN	Vehicular Wireless Sensor Networks
W3C	World Wide Web Consortium
WSN	Wireless Sensor Networks
XML	Extensible Markup Language
ZTL	Zona a Traffico Limitato

2.2 Documenti di riferimento

2.2.1 Standards e normative applicabili

[Std 1]	TBD

2.2.2 Documenti Sii-Mobility

[SiiM 1]	DE1.1a v3-0 - Analisi dei requisiti e casi d'uso
[SiiM 2]	DE1.2a v4-5 - Specifica di Integrazione e Casi di Test
[SiiM 3]	DE8.5 v0-6- Manuale di qualità del progetto
[SiiM 4]	DE3.6a v8 - Sensori: specifica di dettaglio dello sviluppo
[SiiM 5]	DE3.11a v0-11 - Attuatori integrati - specifica di dettaglio dello sviluppo
[SiiM 6]	Sii-Mobility-meeting-2016-11-28-v7-0 (PPT), verbale della riunione
[SiiM 7]	2016.09.16 Verbale riunione skype OR3
[SiiM 8]	2017.02.02 Verbale riunioni skype OR3 kit bike
[SiiM 9]	DE3.1a v0-22 – Kit veicolari - specifica di dettaglio dello sviluppo
[SiiM 10]	DE3.4 - Kit veicolari: manuale di installazione e programmazione
[SiiM 11]	DE3.2 - Kit veicolari: prototipi
[SiiM 12]	DE3.3 - Kit veicolari: rapporto di sperimentazione
[SiiM 13]	DE3.6b - Sensori: specifica di dettaglio dello sviluppo

2.2.3 Altri riferimenti

[Ref 1]	Arduino Board Open Source (http://www.arduino.org/)
[Ref 2]	Arduino 101 (https://www.arduino.cc/en/Main/ArduinoBoard101)
[Ref 3]	Manuale installazione TURMS

2.3 Contesto

Questo documento è prodotto nel contesto del Obiettivo Realizzativo *OR3 - Sviluppo di prototipi applicativi verticali, sensori e attuatori* e in particolare della *Attività 3.1- Studio, definizione e sviluppo di Kit per veicoli: bici, auto e mezzi pubblici e privati.*

Il documento è conforme alle prescrizioni del manuale della qualità del progetto ([SiiM 3]).

La *Figura 1* mostra il ruolo dei Kit veicolari nell'architettura generale di Sii-Mobility. Complessivamente, i kit veicolari hanno lo scopo di:

- interagire con sensori e attuatori presenti sul veicolo per acquisirne i dati e comandi/monitorarli (quando questo è possibile),
- ospitare su smartphone o tablet di servizio Applicazioni Mobili (*Mobile App*) contenenti *Data Harvester* per la raccolta dei dati, fornendo loro i servizi per svolgere le loro funzioni, includendo anche le azioni descritte al punto precedente,
- Mobile App (contenenti Data Harvester) sono installate sui Kit veicolari scaricandole dal Pool di Moduli reso disponibile dalla piattaforma Sii-Mobility. Interagiscono con i sistemi informativi dei Gestori di veicoli oppure direttamente con la piattaforma Sii-Mobility.

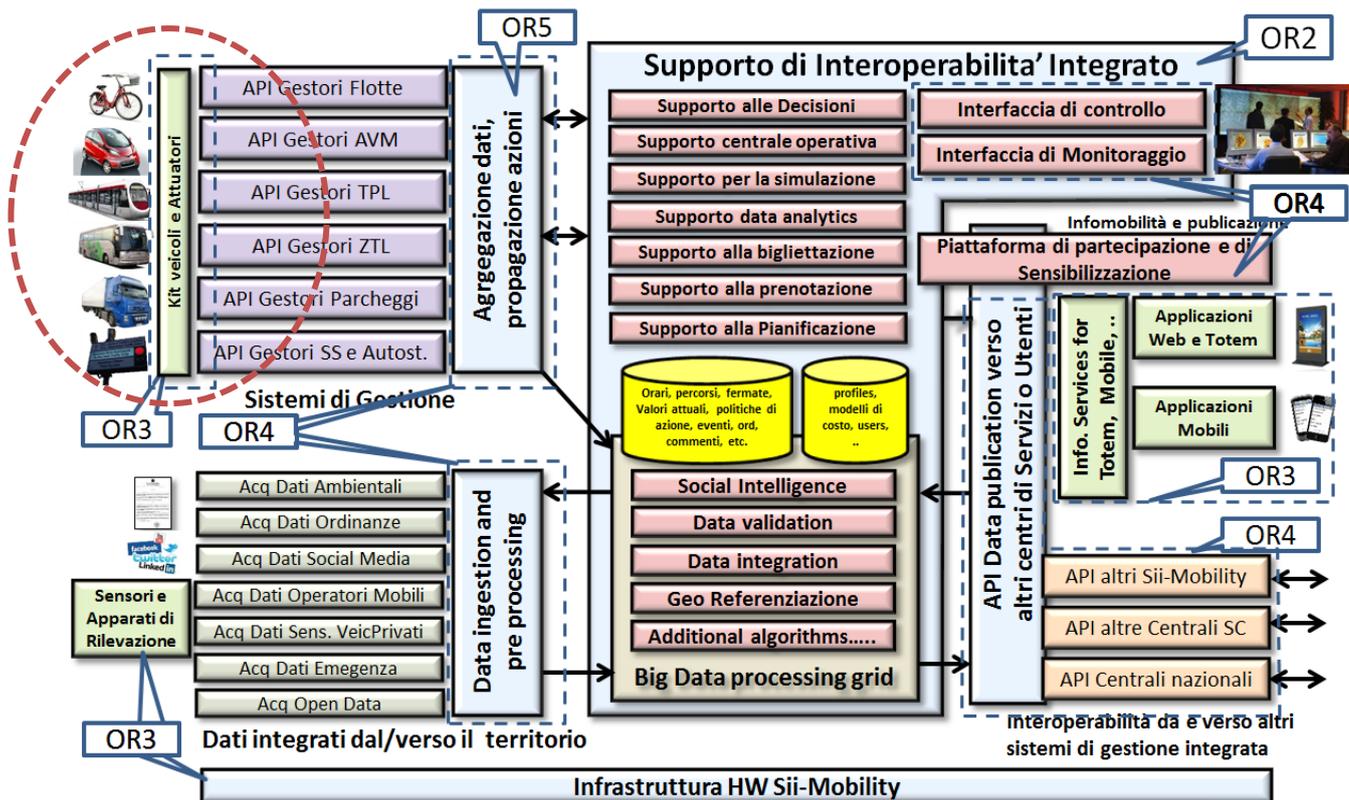


Figura 1 : i Kit veicolari nel contesto di Sii-Mobility

2.4 Notazioni

2.4.1 Schemi di architettura

Negli schemi di architettura generale è utilizzata la notazione in *Figura 2* per rappresentare i flussi di dati e controlli.

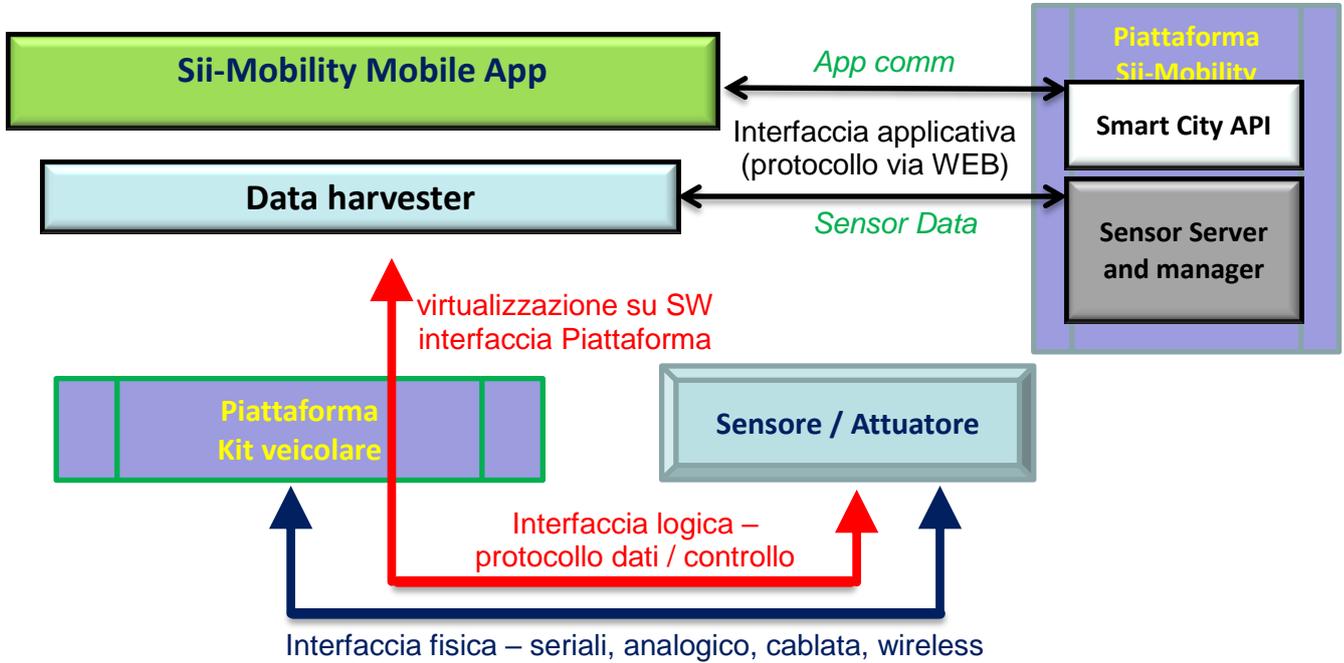


Figura 2 : notazione grafica generale

2.4.2 Schemi di dettaglio

Negli schemi di dettaglio è utilizzata una notazione semi-formale descritta in *Figura 3* ispirata ai diagrammi di Gane-Sarson, a loro volta derivati dalla notazione DFD (Data Flow Diagram).

Interfaccia, elemento esterno		Processo, attività	
Database, archivio, file		Flusso di dati	

Figura 3 : notazione grafica di dettaglio

Negli schemi si rispetta la convenzione sui colori dei flussi dati descritta alla sezione precedente, utilizzando il verde per quei flussi di dati che comprendono più casi di interfacciamento (p.es. sia fisico che logico).

3 Kit bike

Il Kit bike è progettato per equipaggiare biciclette (preferibilmente elettriche) di proprietà di gestori, di noleggi oppure di privati cittadini.

Il Kit include un apparato HW/SW progettato e sviluppato appositamente per il progetto, che nel seguito viene riferito come *Piattaforma*. Si interfaccia fisicamente con *Sensori e Attuatori* installati sul veicolo (e eventualmente ne contiene altri).

In ogni caso, l'Utente deve necessariamente utilizzare il proprio *Smartphone* per accedere ai servizi di Sii-Mobility: lo smartphone viene quindi utilizzato come parte integrante del Kit e su di esso viene scaricata la Mobile App (contenente il Data Harvester)

La Piattaforma e lo Smartphone comunicano utilizzando gli apparati Bluetooth Low Energy di cui entrambi sono equipaggiati. Realizzano complessivamente un meccanismo di comunicazione che permette uno scambio di comandi / controlli / informazioni in modalità wireless.

3.1 Architettura generale

L'architettura generale, contestualizzata in Sii-Mobility, del Kit bike è mostrata in *Figura 4*

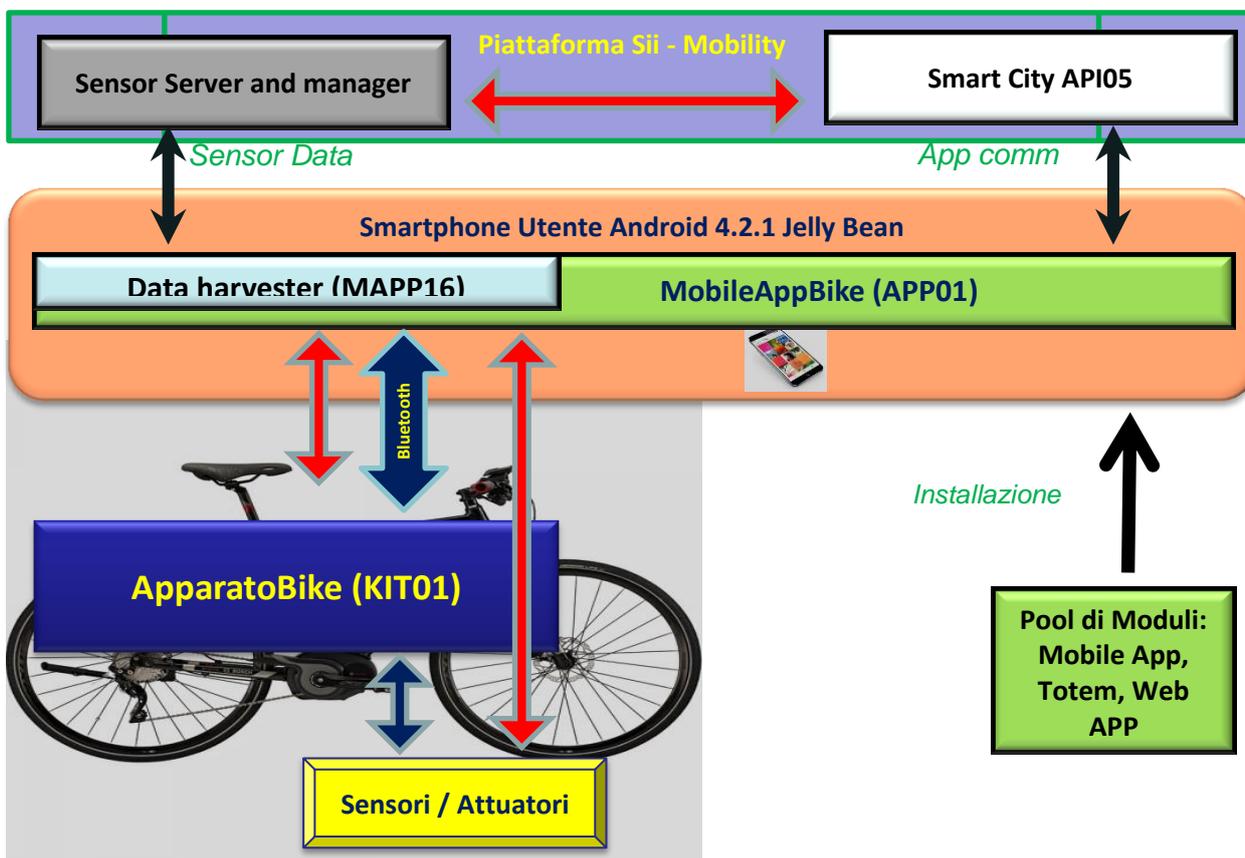


Figura 4 : Architettura generale del Kit bike

I vantaggi offerti da questa organizzazione sono molteplici e rilevanti:

- chi sviluppa la Mobile App (contenente il Data Harvester) opera in un ambiente “familiare”, costituito da uno smartphone commerciale, con caratteristiche ben standardizzate. Utilizza una installazione originale del sistema operativo Android, mentre un “porting” di Android sulla Piattaforma esporrebbe a rischi di comportamenti anomali,

- il canale di comunicazione Bluetooth può essere usato efficacemente sia per rendere disponibili al Data Harvester i dati dei sensori contenuti nella piattaforma stessa, sia per virtualizzare l'interazione con i sensori esterni alla piattaforma,
- la Piattaforma risulta essere sostanzialmente una estensione del Data Harvester, limitando di fatto il numero di partners coinvolti nella definizione e implementazione delle interfacce tra Data Harvester e Piattaforma e tra Data Harvester e sensori,
- la Piattaforma si basa su una architettura hardware “open” che costituisce uno *standard de facto* per applicazioni di questo tipo ([Ref 1]).

Il Kit Bike include apparati, sensori e attuatori a supporto di una funzione anti furto (ATF, anti-theft feature). Si veda la sezione 3.2.3 per dettagli.

3.2 Specifica di dettaglio

3.2.1 Diagramma di contesto generale

La *Figura 5* mostra un primo livello di specifica e ripropone lo schema generale di architettura nella notazione utilizzata per la descrizione di dettaglio. A questo livello occorre definire con maggior precisione le interfacce mostrate.

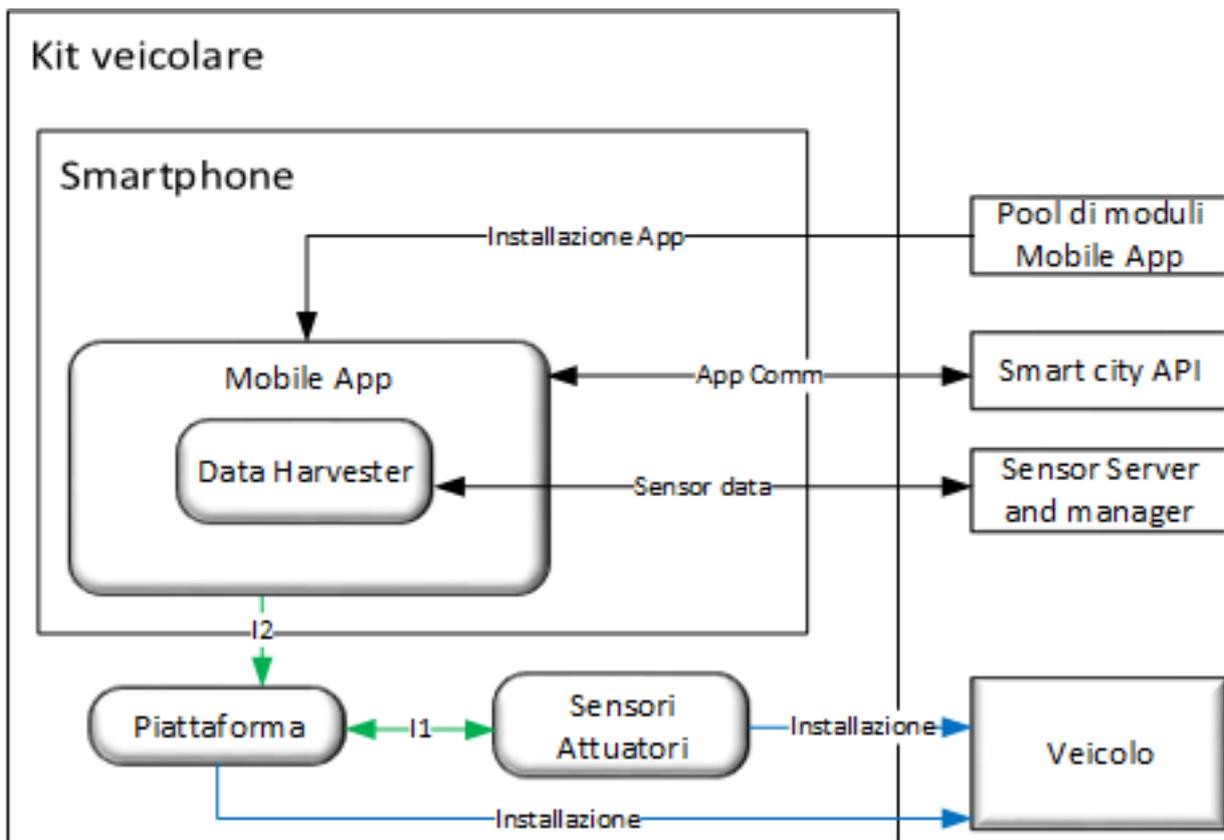


Figura 5 : Kit Bike - diagramma di contesto

In particolare:

- Interfacce esterne:

- *Installazione App, App Comm e Sensor data* sono interfacce applicative realizzate con scambi di dati su WEB. Non vengono ulteriormente dettagliate in questo documento;
- *Installazione* della Piattaforma e dei Sensori è un'interfaccia fisica che coinvolge la collocazione dei vari elementi. Richiede un momento di coordinamento tra i partners che realizzano i contenitori ed i supporti dei vari elementi del Kit e quelli che si occupano della installazione sui veicoli, propedeutica alla sperimentazione. L'argomento è sviluppato alla sezione 3.2.2.5;
- Interfacce interne, sviluppate nelle sezioni che seguono:
 - I1 è l'interfaccia fisica e logica tra Piattaforma e Sensori;
 - I2 è l'interfaccia fisica e logica tra Data Harvester e Piattaforma;

3.2.2 Piattaforma, KIT01

3.2.2.1 Diagramma di contesto

In *Figura 6* il diagramma di contesto della Piattaforma. Si fa distinzione tra sensori / attuatori:

- *passivi*, per i quali l'interfaccia fisica è costituita da collegamenti elettrici e l'interfaccia logica dai livelli di tensione presenti sui collegamenti;
- *attivi*, che includono una CPU, per i quali l'interfaccia fisica è costituita da linee seriali e l'interfaccia logica dai messaggi scambiati secondo un certo protocollo.

I sensori / attuatori attivi potranno anche avere collegamenti di comando / controllo analoghi a quelli dei sensori passivi in aggiunta allo scambio di messaggi.

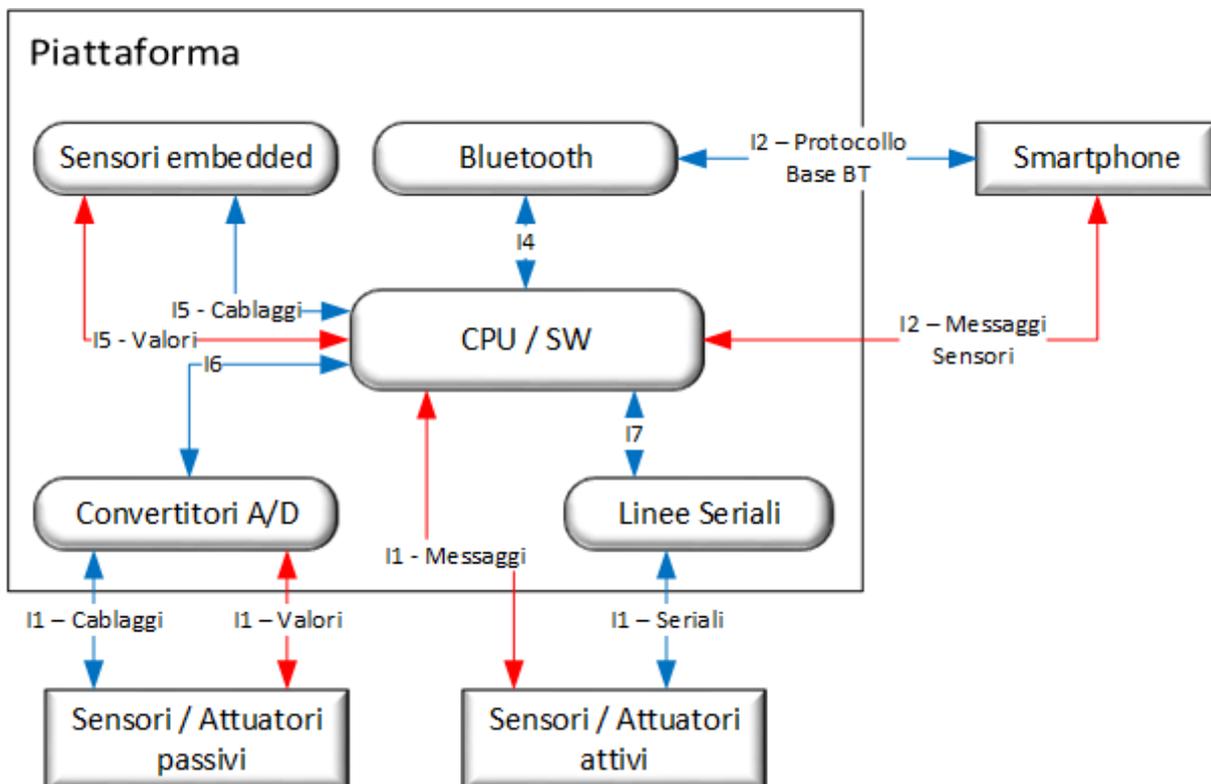


Figura 6 : Piattaforma – diagramma di contesto

La piattaforma include gli elementi seguenti, tutti descritti in dettaglio alla sezione 3.2.2.3:

- convertitori Analogico/Digitale e Digitale/Analogico per la gestione dei collegamenti elettrici con i sensori e attuatori passivi e attivi. Hanno riscontro nelle porte di I/O della motherboard;
- linee seriali per lo scambio di messaggi con i sensori e attuatori attivi. Hanno riscontro nelle porte di I/O della motherboard;
- una scheda CPU (motherboard) che include:
 - un dispositivo Bluetooth Low Energy per le comunicazioni con lo smartphone;
 - un modulo accelerometro/giroscopio a 6 assi in grado di fornire dati per eseguire analisi sullo stato della strada e sulla condotta del veicolo.

I sensori / attuatori interfacciati sono:

- un apparato GPS per supportare la funzione antifurto. È descritto in dettaglio alla sezione 3.2.5.2;
- un segnale sonoro di allarme per supportare la funzione antifurto e la funzione anticollisione. È descritto in dettaglio alla sezione 3.2.5.3;
- sensori ambientali, descritti in dettaglio alla sezione 3.2.5.4;
- sensori anti-collisione, descritti in dettaglio alla sezione 3.2.5.5;

Principali attività e interfacce coinvolte:

- ❖ **gestione sensore / attuatore passivo:** i dati dai sensori embedded sono acquisiti e processati direttamente sulla piattaforma. *I1 – Cablaggi* rappresenta i collegamenti cablati con il sensore / attuatore mentre *I1 – Valori* rappresenta il significato della grandezza (tipicamente una tensione) che si acquisisce o si attua per ciascun collegamento, p.es. proporzionale alle parti per milione di un agente inquinante, oppure proporzionale alla distanza del veicolo che segue. La piattaforma acquisisce i valori, li codifica in messaggi secondo le convenzioni definite per *I2 – protocollo Base BT* e li invia allo smartphone (Data Harvester). È importante notare che viene trasmesso il dato grezzo, il suo significato (interfaccia logica) non è rilevante per la piattaforma, mentre lo è per il Data Harvester. L'interfaccia *I6* rappresenta le convenzioni, locali alla piattaforma (ECM-ELFI), sulla locazione dei convertitori e sulla loro gestione;
- ❖ **gestione sensore / attuatore attivo:** i messaggi dai sensori (*I1 - Messaggi*) sono ritrasmessi senza modifiche allo smartphone e viceversa secondo le convenzioni definite per *I2 – protocollo Base BT*. L'interfaccia *I1 – Seriali* rappresenta solo il tipo di linea seriale, mentre *I7* rappresenta le convenzioni, locali alla piattaforma (ECM-ELFI), sulla locazione dei buffer e sulla gestione degli eventi TX/RX. Se un sensore / attuatore attivo ha anche collegamenti cablati, questi sono gestiti come nel caso precedente;
- ❖ **le interfacce fisiche** con i sensori / attuatori sono condivise tra chi sviluppa il sensore / attuatore (che le definisce) e chi sviluppa l'HW della piattaforma (ELFI) che le recepisce;
- ❖ **le interfacce logiche** con i sensori / attuatori sono condivise tra chi sviluppa il sensore / attuatore (che le definisce) e chi sviluppa il Data Harvester (DISIT) che le recepisce;
- ❖ **gestione del collegamento Bluetooth :** il SW della Piattaforma e quello del Data Harvester definiscono e condividono le informazioni necessarie per realizzare un canale di comunicazione Bluetooth (interfaccia *I2 – protocollo Base BT*) con un suo protocollo. Utilizzano, rispettivamente, i servizi della piattaforma, interfaccia *I4*, e di Android (vedere *Figura 11*). Il protocollo comprende il formato dei messaggi (header, footer, etc.), messaggi di controllo e messaggi di dati. Nei messaggi dati sono contenuti (senza modifiche) i

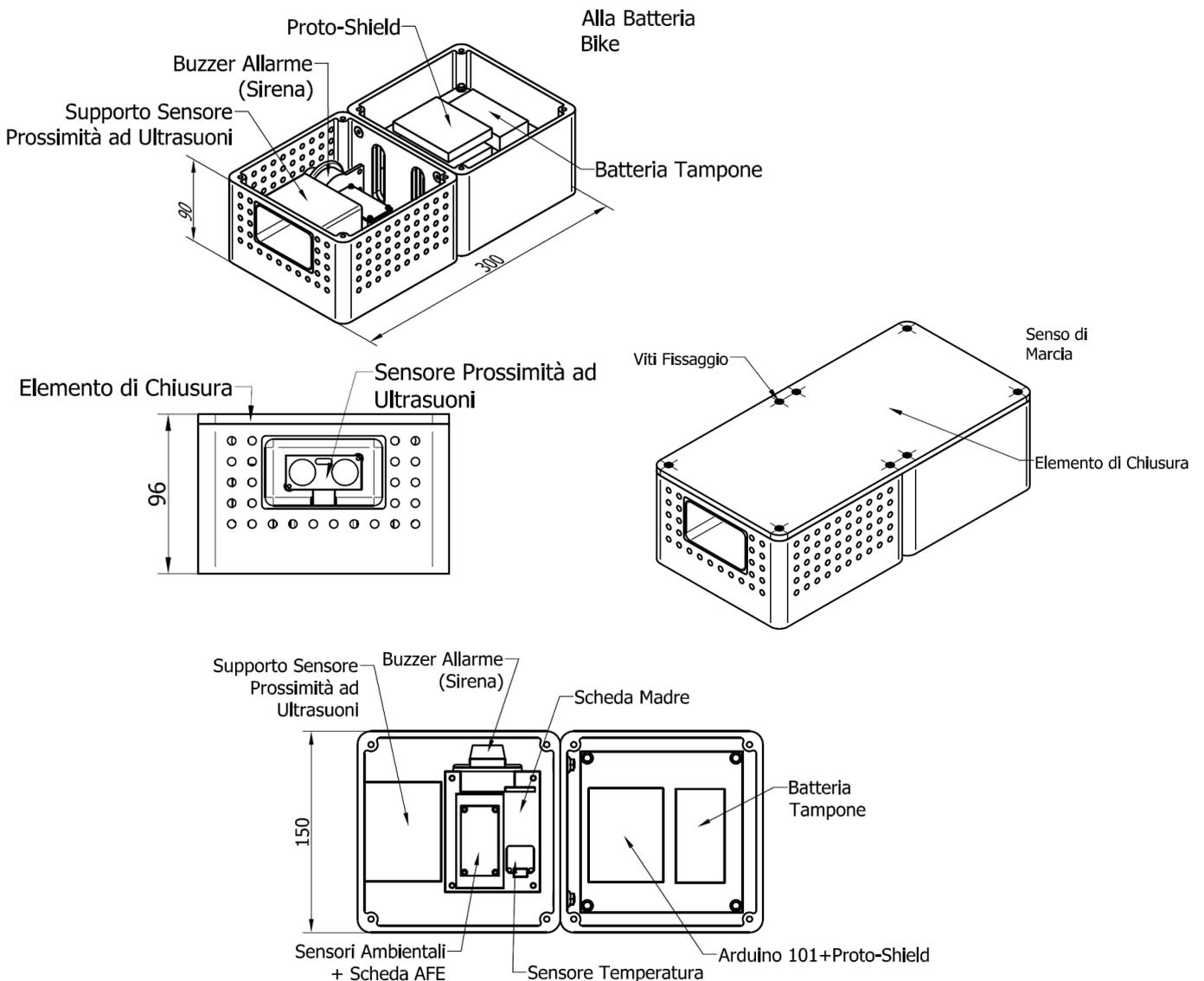
messaggi da e per sensori / attuatori attivi e inoltre i messaggi generati dalla piattaforma per i sensori / attuatori passivi.

Dunque l'interfaccia logica *I2 – Messaggi sensori* è costituita dall'OR di:

- *I1 – Valori*
- *I1 - Messaggi*
- *I5 – Valori*

3.2.2.2 Meccanica, contenitori, carpenterie

Con riferimento ai componenti ad oggi individuati, è stata studiata una prima soluzione per il contenitore. Questa prevede una box in materiale plastico, di dimensioni compatte, che contiene tutto il kit bike. È divisa in due sezioni contigue: una a tenuta di acqua e di polvere, l'altra dotata di fori di aerazione per consentire ai sensori installati (sensori ambientali, di temperatura, di



prossimità) il contatto con l'aria dell'ambiente esterno. Lo schema è riportato in Figura 7.

Figura 7 : Kit bike – Contenitore e disposizione apparati

Il contenitore è stato pensato per essere installato sul portapacchi della bici, sopra la ruota posteriore.

3.2.2.3 Hardware

3.2.2.3.1 Motherboard

Come scheda madre viene individuata la scheda Arduino 101 (Genuino 101 in Europa), di recente realizzazione, che dispone di un modulo processore Intel Curie e integra un modulo Bluetooth Low Energy (BLE) e un modulo accelerometro/giroscopio a 6 assi.

La piattaforma si presenta con le stesse dimensioni, lo stesso pin-out e periferiche di Arduino Uno (rendendo le shields sviluppate per Arduino Uno compatibili con il 101), ma con architettura sensibilmente diversa. I pin I/O lavorano a 3.3V (rispetto ai 5V di Arduino uno), anche se protetti rispetto ai 5V.

La scheda si basa sul modulo Intel Curie, che contiene due processori, un x86 (Intel Quark) e un 32-bit ARC, entrambi a 32MHZ, diversamente da Arduino uno, basato su ATmega328, fornendo sulla carta un miglioramento delle prestazioni e un minor consumo.

Sul processore Intel Quark gira un sistema operativo RTOS, ancora in fase di sviluppo e per il quale sono previsti periodici aggiornamenti; mentre il processore ARC è dedicato a far girare i programmi di Arduino e a prendersi cura dell'I/O. I due processori operano simultaneamente e condividono la memoria; in questo modo sono disponibili per uso del software esterno 196kB (dei 384 kB) di flash memory, e di 24kB (degli 80 kB) di SRAM.

I due processori comunicano attraverso static mailboxes, con particolare riferimento a alcune operazioni (interfaccia attraverso porta USB; caricare il programma in flash; gestire l'interazione con il Bluetooth; generare PWM).

Il modulo Intel Curie è stato sviluppato in particolare per applicazioni wearable, con particolare attenzione ai bassi consumi, alla connettività (Bluetooth Low Energy) e con il sensore accelerometrico/giroscopio che rende semplice sviluppare analisi del movimento.

La combinazione di un accelerometro e un di un giroscopio inclusi nella piattaforma costituisce una Unità di Misura Inerziale (Inertial Measurement Unit o IMU), capace di identificare orientazione e movimento dell'oggetto. Una specifica libreria (Curie IMU) permette la lettura e l'elaborazione dei dati grezzi provenienti dal giroscopio e dall'accelerometro.

L'Arduino 101 ha 14 piedi digitali di input/output (dei quali 4 usabili come PWM), 6 piedi di input analogici, con connettore USB per la comunicazione seriale e per il caricamento degli sketches di Arduino, un connettore ICSP con segnali SPI, e piedi dedicati alla comunicazione I2C. Tutti i piedi operano a 3.3 volt e possono essere usati come interrupt.

Per ulteriori dettagli si veda ([Ref 3]).

3.2.2.3.2 Bluetooth Low Energy

Per quanto riguarda la connettività il protocollo Bluetooth Low Energy (o Bluetooth Smart, parte dello sviluppo Bluetooth 4.0) è stato sviluppato nell'ottica del basso costo e del basso consumo, a parità di distanza di comunicazione, per applicazioni di tipo IoT, ovvero con bassi data rate (non adatto per esempio a streaming audio o video). I sistemi operativi mobile e non, tra cui iOS, Android, BlackBerry, macOS, Linux, Windows8 e 10, lo supportano in modo nativo e tra i vantaggi di questo protocollo c'è quello di poter realmente interagire con una grande varietà di piattaforme mobili, telefoni, tablet e computer.

Il data-rate teorico del BLE è di 1 Mb/s, ma in realtà il massimo data rate realmente raggiungibile è intorno ai 10kB/s, con sensibili differenze a seconda del sistema operativo che lo utilizza (con un throughput da meno di 3kB/s a 13 kB/s circa).

Diversamente dal Bluetooth standard basato su connessione seriale asincrona (UART), il Bluetooth LE agisce in modo diverso. Esistono due funzioni principali che ciascun elemento può compiere: pubblicare notizie (e in questo caso si chiama “peripheral device”, che agisce come un server) o leggerle (“central device”, che agisce come client). Più central devices possono accedere alle notizie pubblicate da una singola periferica; queste ultime sono presentate come servizi (che possono essere standard o personalizzati), e provvisti di identificativo numerico univoco (UUID), ciascuno suddiviso in caratteristiche. Un meccanismo di notifica informa quando i dati sono cambiati. Questa struttura è chiamata publish-and-subscribe model (si tende a notificare solo quando ci sono cambiamenti e non su intervalli di tempo regolari). I Peripheral devices forniscono per esempio come caratteristiche i dati di un sensore, oppure permettono di leggere/scrivere su alcune caratteristiche per comandare attuatori.

Il valore di ciascuna caratteristica può essere lunga al massimo 20 byte: questo è un vincolo cruciale del protocollo.

Un central device (che agisce come client) ha 4 funzioni rispetto a una caratteristica: lettura (chiedere alla periferica di fornire il valore della caratteristica), scrittura (modificarne il valore, come con gli attuatori), indicazione e notifica (chiedere alla periferica di mandare in modo continuo i valori aggiornati della caratteristica, senza chiederli ogni volta).

GAP (General Advertising Profile) è il modo in cui ciascun dispositivo BLE rende nota la sua esistenza; mentre il GATT (General ATtribute Profile) definisce servizi e caratteristiche e abilita le operazioni di lettura/scrittura/notifica.

3.2.2.3.3 Piattaforma inerziale embedded

La scheda Arduino 101 è dotata della Intel Curie IMU costituita da giroscopio e accelerometri triassiali con i quali stimare sia la posizione angolare della scheda che le accelerazioni a cui è soggetta. Vengono messe a disposizione librerie che offrono la possibilità di calibrare i sensori accelerometrici e giroscopici una volta posizionata la scheda sul veicolo. I sensori hanno dei parametri settabili come la velocità di acquisizione e il massimo range misurabile come indicato in Tabella 1.

Parametro	Accelerometro	Giroscopio
Range (g ; °/s)	<ul style="list-style-type: none"> • 2 (+/- 2g) • 4 (+/- 4g) • 8 (+/- 8g) • 16 (+/- 16g) 	<ul style="list-style-type: none"> • 2000 (+/-2000°/s) • 1000 (+/-1000°/s) • 500 (+/-500°/s) • 250 (+/-250°/s) • 125 (+/-125°/s)

Frequenza di acquisizione (Hz)	• 12.5	• 25
	• 25	• 50
	• 50	• 100
	• 100	• 200
	• 200	• 400
	• 400	• 800
	• 800	• 1600
	• 1600	• 3200

Tabella 1: Intel Curie IMU-Parametri settabili dei sensori accelerometrico e giroscopico

3.2.2.3.4 Interfacciamento dei sensori / attuatori

La motherboard interagisce con i sensori descritti nelle sezioni seguenti: ovvero un sensore di distanza a ultrasuoni (3.2.5.5), sensori ambientali (3.2.5.4), un modulo GPS (3.2.5.2). È inoltre necessario un sensore di temperatura/umidità (3.2.5.5) per la correzione dei dati del sensore a ultrasuoni.

I sensori ambientali richiedono dell'hardware per l'adattamento dei segnali, contenuto in un Modulo di condizionamento. Ci si propone di realizzare una proto-shield aggiuntiva dove posizionare detto HW e inoltre i convertitori 5/3.3 volt, l'HW di gestione del segnale di allarme e la scheda GPS.

Le alimentazioni dei sensori saranno fornite tutte da linea esterna e non dalle porte della piattaforma: questa è una soluzione che garantisce maggiore robustezza e diminuisce il rischio di interferenze nel funzionamento di tutto il kit. La Figura 8 e la Tabella 2 mostrano uno schema generale del kit e il dettaglio del collegamento delle interfacce fisiche alle porte di Arduino 101.

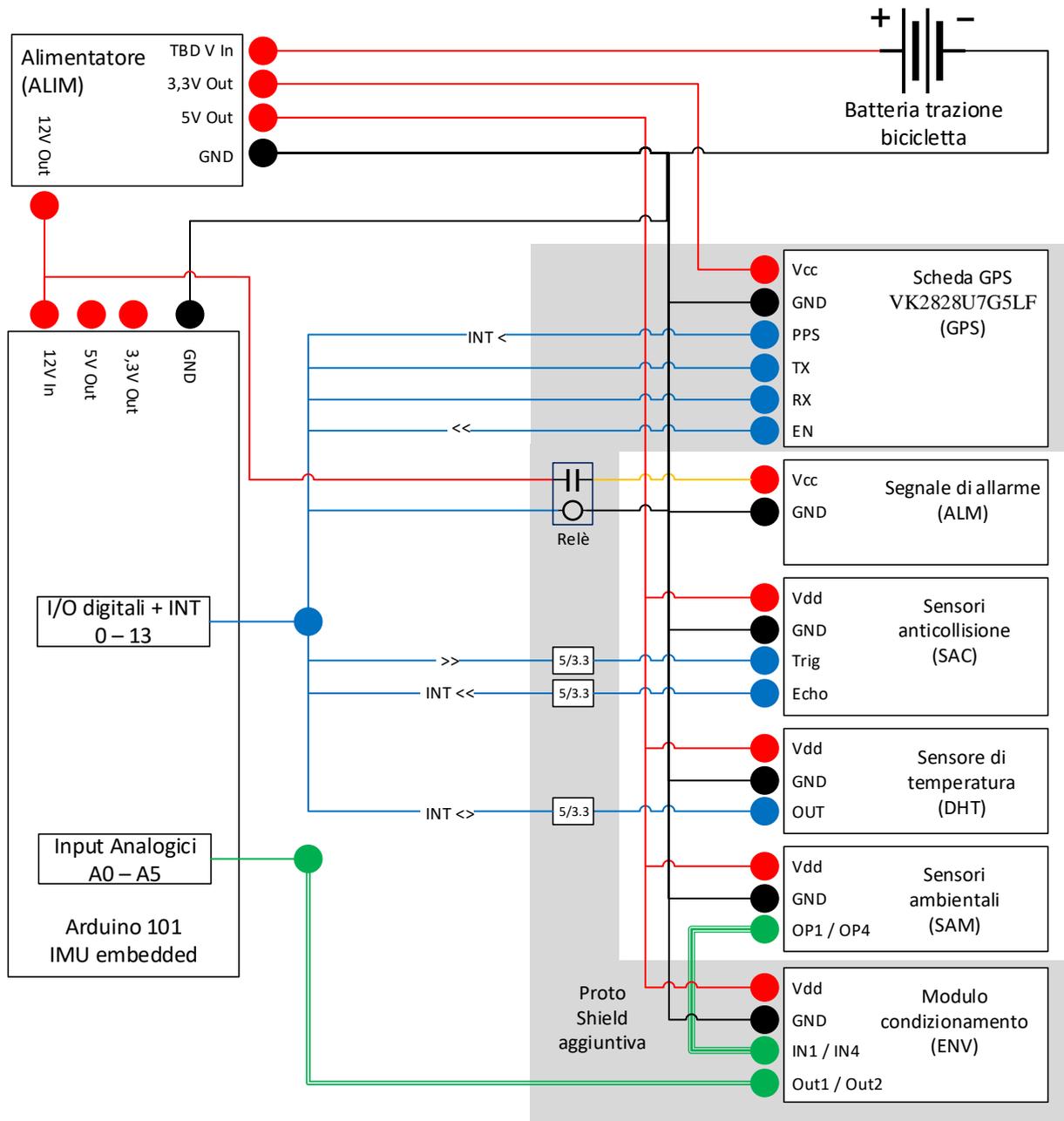


Figura 8 : Schema generale del kit bike

Kit bike - Connessioni fisiche tra piattaforma e sensori							
Analog IN	A0	A1	A2	A3	A4	A5	
	ENV Out1	ENV Out2					
I/O/INT Ports	0/RX	1/TX	2°	3*	4	5*°	6*°

	GPS RX	GPS TX	GPS PPS	GPS EN	ALM		
I/O/INT Ports	7	8°	9*	10°	11°	12°	13°
	SAC Trig	SAC Echo	DHT OUT				
Power	ATN/SS	IOREF	RESET	3,3	5	GND	Vin
						Ground	12V
Other	SCL	SDA	AREF	ICSP/ MISO	ICSP/ MOSI	ICSP/ SCK	ICSP

Tabella 2: Kit bike - connessioni fisiche tra motherboard e sensori

*: porte che supportano PWM

°: porte che supportano interrupt sui cambiamenti di stato

3.2.2.4 Software

In questo paragrafo è descritto il design del firmware della Piattaforma, denominato Vehicular Kit Manager Bike (VKMbike).

3.2.2.4.1 Funzioni del SW VKMbike

La Piattaforma è dotata di firmware che implementa la logica di gestione dei sensori installati e/o collegati alla Piattaforma e la logica di gestione della comunicazione Bluetooth con lo Smartphone. Il VKMbike è in grado di:

- Acquisire il dato di misura (grezzo) dei sensori installati e/o collegati alla Piattaforma – I1;
- Gestire i dati di misura eventualmente secondo una logica basata su priorità;
- Realizzare Protocollo Base BT (si veda la Sezione 0 per dettagli);
- Inviare i dati acquisiti al Data Harvester – I2
- Comandare / controllare i sensori / attuatori secondo i dati di configurazione di Piattaforma e / o su comando del Data Harvester

Al fine di fornire servizi a valore aggiunto all'utente e/o al gestore di noleggio bici, il VKMbike implementa una funzionalità di segnalamento di primaria importanza per la funzione Autonomous Anti-Theft Feature (AATF) descritta nella sezione 3.2.3.

3.2.2.4.2 Casi d'uso

In Figura 9 è mostrato il diagramma UML dei casi d'uso in cui il VKMbike è coinvolto da parte degli attori che si interfacciano con esso. Si è assunto che l'utente possa decidere se utilizzare un servizio di bike sharing con biciclette equipaggiate con Kit Bike, oppure di procurarsi in modo autonomo il Kit Bike che installerà sulla propria bicicletta. Independentemente dalla scelta dell'utente, si hanno tuttavia i seguenti casi d'uso.

- **Vehicular association:** coinvolge l'interfaccia I2. È necessario per l'attivazione degli altri casi d'uso e per usufruire del servizio ATF;

- **ID:** UC1;
- **Attori:** Smartphone, Piattaforma;
- **Pre-condizione:**
 1. Lo Smartphone è attivo con installate la Mobile App ed il Data Harvester;
 2. La Piattaforma è funzionante;
 3. Il modulo Bluetooth della Piattaforma e dello Smartphone è attivo;
- **Sequenza degli eventi:** secondo il Protocollo Base BT (si veda la Sezione 3.2.6 per dettagli)
 1. Il caso d'uso inizia quando la Piattaforma riceve un messaggio di associazione da parte dello Smartphone;
 2. La Piattaforma invia allo Smartphone un messaggio di risposta alla richiesta di associazione;
- **Post-condizione:** lo Smartphone e la Piattaforma sono associati e sono attivati gli altri casi d'uso;
- **Data gathering:** coinvolge l'interfaccia I1. Acquisizione periodica dei dati di misura dei sensori:
 - **ID:** UC2;
 - **Attori:** Piattaforma, Time;
 - **Pre-condizione:** UC1 completato;
 - **Sequenza di eventi:**
 1. Il caso d'uso inizia quando scade il Timer periodico di gathering;
 2. La Piattaforma legge il valore di tensione del sensore;
 3. La Piattaforma esegue l'operazione di Timestamping e georeferenziazione della misura;
 4. La Piattaforma imposta il Timer periodico di gathering;
 - **Post-condizione:** la Piattaforma ha acquisito il dato del sensore;
- **Data sending:** coinvolge l'interfaccia I2. Invio dei dati allo Smartphone (Data Harvester):
 - **ID:** UC3;
 - **Attori:** Piattaforma, Time
 - **Pre-condizione:** UC1 completato;
 - **Sequenza di eventi:** secondo il Protocollo Base BT (si veda la Sezione 3.2.6 per dettagli)
 1. Il caso d'uso inizia quando scade il Timer periodico di data sending;
 2. La Piattaforma invia allo Smartphone un messaggio contenente i dati di misura dei sensori acquisiti nel caso d'uso UC2;
 3. La Piattaforma imposta nuovamente il Timer periodico di data sending;
 - **Post-condizione:** La Piattaforma ha inviato il messaggio ed il Timer periodico è attivo.
- **Beaconing:** coinvolge l'interfaccia I2. Invio di messaggi di segnalazione, detti beacon, necessari per l'implementazione del IAT:
 - **ID:** UC4;
 - **Attori:** Piattaforma, Time;
 - **Pre-condizione:** UC1 completato;
 - **Sequenza di eventi:** secondo il Protocollo Base BT (si veda la Sezione 3.2.6 per dettagli)
 1. Il caso d'uso inizia quando scade il Timer periodico di beaconing;
 2. La Piattaforma invia allo Smartphone un beacon;
 3. La Piattaforma imposta nuovamente il Timer periodico di beaconing;
 - **Post-condizione:** la Piattaforma ha inviato il messaggio beacon ed il Timer periodico è attivo.
- **Clock synchronization:** coinvolge l'interfaccia I2. Sincronizzazione dei clock tra Piattaforma e Smartphone, dettagli in Sezione 3.2.6.2 Sincronizzazione del Clock di Piattaforma;
 - **ID:** UC5
 - **Attori:** Piattaforma, Time, Smartphone
 - **Pre-condizione:** UC1 completato;
 - **Sequenza di eventi:** secondo il Protocollo Base BT (si veda la Sezione 3.2.6 per dettagli)

1. Il caso d'uso inizia quando scade il Timer di sincronizzazione;
 2. La Piattaforma invia allo Smartphone un messaggio di clock synchronization;
 3. Lo Smartphone risponde con il clock;
 4. La Piattaforma imposta il Timer di sincronizzazione;
- **Post-condizione:** i clock sono sincronizzati
- **Anti-Theft:** è il caso d'uso dello scenario autonomous anti-theft feature (AATF)
 - **ID:** UC6
 - **Attori:** Piattaforma
 - **Pre-condizione:** la Piattaforma è attiva
 - **Sequenza di eventi:**
 1. Il caso d'uso inizia quando termina il caso d'uso UC7
 2. La Piattaforma inizia ad eseguire il tracking GPS
 - Se la Piattaforma rileva, secondo un opportuno meccanismo, uno spostamento indebito attiva un segnale sonoro
 - **Post-condizione:** la Piattaforma emette un segnale sonoro
 - **Vehicular deassociation:** deassociazione tra Piattaforma e Smartphone
 - **ID:** UC7
 - **Attori:** Piattaforma, Smartphone
 - **Pre-condizione:** UC1 completato;
 - **Sequenza di eventi:** secondo il Protocollo Base BT
 1. Il caso d'uso inizia quando la Piattaforma riceve dallo Smartphone un messaggio di deassociazione;
 2. La Piattaforma risponde allo Smartphone
 - **Post-condizione:** Piattaforma e Smartphone sono deassociati

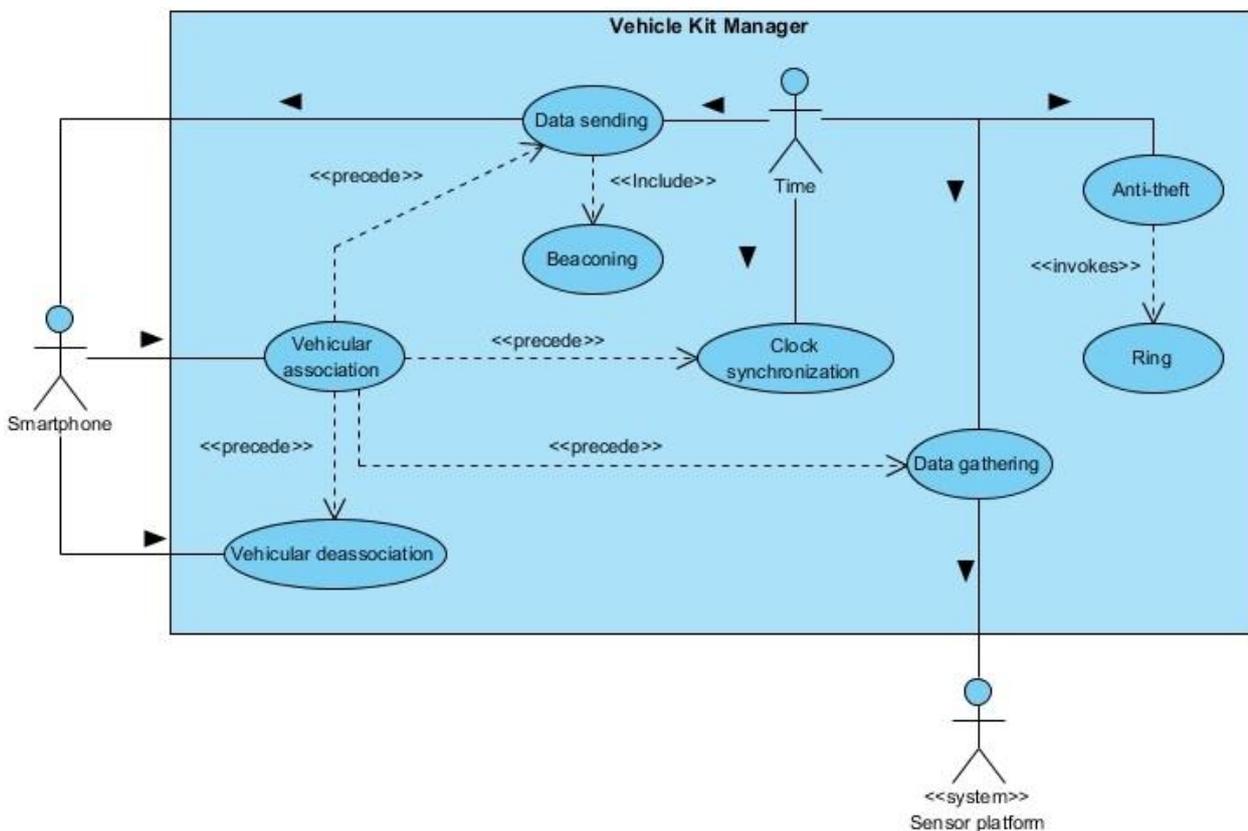


Figura 9 : Vehicular Kit Manager Bike – casi d'uso

3.2.2.4.3 Macchina a stati

Nella seguente Figura 10 è riportata la macchina a stati della Piattaforma secondo il formalismo degli Harel Statechart. All’atto dell’accensione del Kit Bike switchON, il VKMbike esegue una fase preliminare di setup() dell’hardware. Al termine di questa fase preliminare il VKMbike si trova nel superstato Idle.

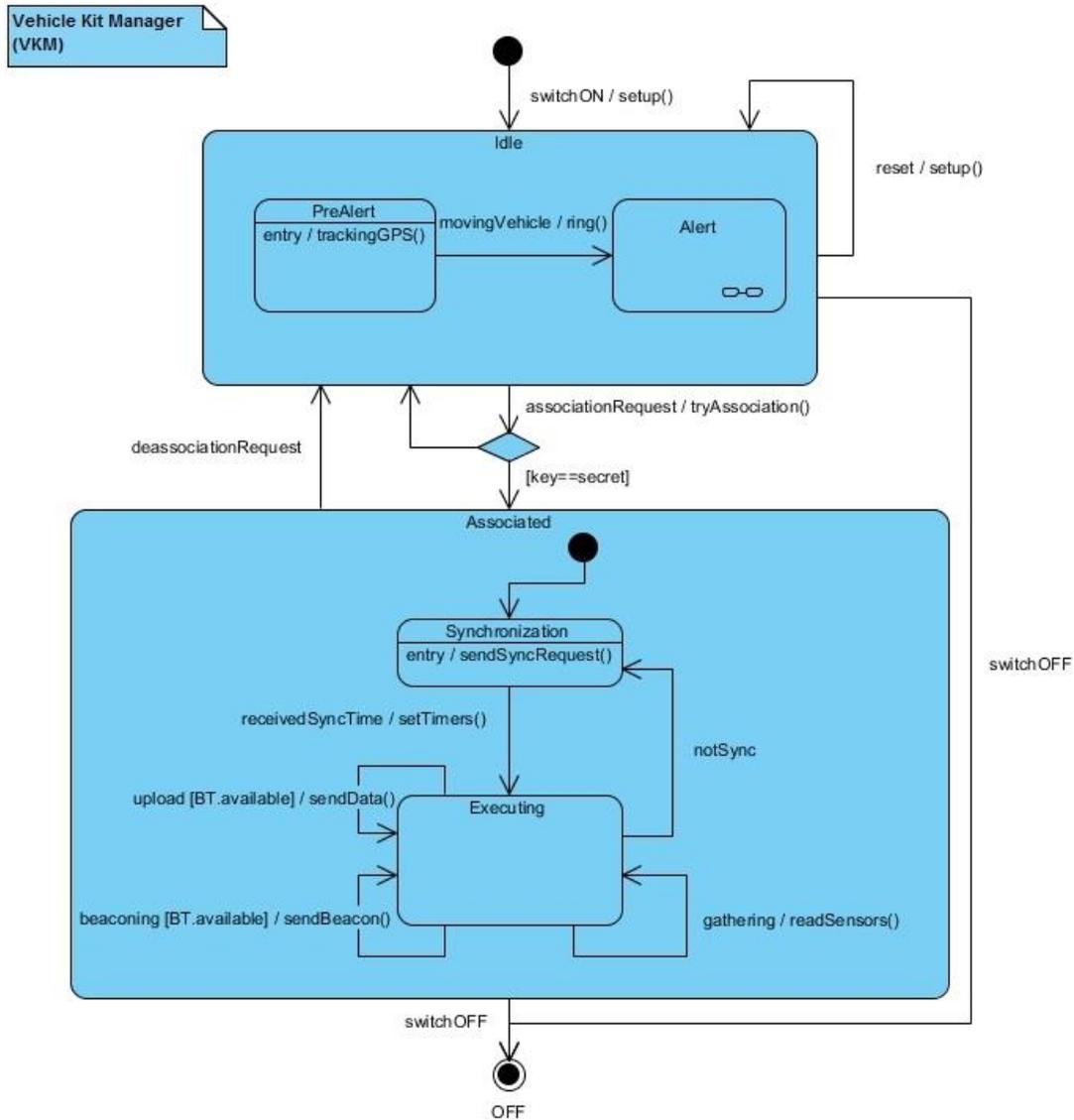


Figura 10: Vehicular Kit Manager Bike – FSM

Descrizione superstato Idle

Il VKMbike è attivo e funzionante, non è associato con nessun dispositivo utente e dunque è in attesa di una richiesta di associazione da parte dello Smartphone.

È attivo lo stato di `Idle.PreAlert`, in cui si esegue l’azione di `trackingGPS()`, che implementa l’AATF, Sezione 3.2.4.2. Da qui se si rileva, secondo un opportuno meccanismo, che il veicolo si sta spostando indebitamente, evento `movingVehicle`, si esegue l’azione `ring()` di allarme. Le azioni in caso di furto sono modellate dalla submachine `Alert`.

Il superstato `Idle` risponde agli eventi di `reset` e `switchOFF`. Se la Piattaforma riceve una richiesta di associazione da parte dello Smartphone, evento `associationRequest`, sono

eseguite le operazioni di associazione. Se la procedura va a buon fine lo stato `Associated` è attivo, viceversa torna attivo lo stato `Idle.PreAlert`.

Descrizione superstato `Associated`

Lo stato attivo è `Associated.Synchronization` in cui sono eseguite le azioni di sincronizzazione del clock di Piattaforma, Sezione 3.2.5.2, `sendSyncRequest()`.

Al verificarsi dell'evento di `receivedSyncTime`, la fase di sincronizzazione è terminata, l'azione `setTimers()` è eseguita e lo stato `Associated.Executing` è attivo. Da qui sono possibili le seguenti transizioni:

- Evento `upload`: invio dei messaggi contenenti i dati di misura dei sensori, `sendData()`, purché sia rilevata attiva la connessione BT;
- Evento `beaconing`: invio dei messaggi di segnalazione per IATF, Sezione 3.2.3.3, azione `sendBeacon()`, purché sia rilevata attiva la connessione BT;
- Evento `gathering`: lettura dei valori di tensione dei sensori, azione `readSensors()`;
- Evento `notSync`: dispositivo non sincronizzato dunque si esegue la transizione che porta allo stato `Synchronization` in cui si rieseguo le operazioni di sincronizzazione dei clock, Sezione 3.2.5.2;

La Piattaforma, quando si trova nel superstato `Associated`, risponde agli eventi di:

- `deassociationRequest` che ha come destinazione il superstato `Idle`, in particolare `Idle.PreAlert`. Questo evento si verifica quando lo Smartphone invia una richiesta di deassociazione;
- `switchOFF`: spegnimento del dispositivo.

3.2.2.4.4 Moduli software

Complessivamente, il software del VKMbike si articola nei moduli:

- **Startup** – esegue l'inizializzazione della Piattaforma e di tutti i sensori collegati;
- **Controllore** – implementa la macchina a stati descritta in 3.2.2.4.3 per realizzare i casi d'uso descritti in 3.2.2.4.2. Attiva i moduli software descritti di seguito con una politica periodica oppure event-driven a seconda dei casi;
- **Gestione Bluetooth** – realizza il comando e controllo dell'hardware di comunicazione Bluetooth;
- **Protocollo base Bluetooth** – implementa, lato Piattaforma, il protocollo di comunicazione con il Data Harvester, (si veda la Sezione 0 per dettagli);
- **Gestione segnale sonoro di allarme** – realizza il comando e controllo dell'attuatore passivo – fornisce servizi agli altri moduli per attivare e disattivare il segnale;
- **Gestione piattaforma accelerometrica embedded** - realizza il comando e controllo del sensore attivo – fornisce al controllore i servizi per l'acquisizione dei dati;
- **Gestione GPS** - realizza il comando e controllo del sensore attivo – fornisce al controllore i servizi per l'acquisizione dei dati;
- **Gestione sensori ambientali** - realizza il comando e controllo dei sensori passivi – fornisce al controllore i servizi per l'acquisizione dei dati;

- **Gestione sensori anti-collisione** - realizza il comando e controllo dei sensori attivi – fornisce al controllore i servizi per l’acquisizione dei dati;
- **Gestione linee seriali** – modulo di servizio per tutti gli altri moduli;
- **Gestione segnali analogici** – modulo di servizio per tutti gli altri moduli.

3.2.2.5 Installazione a bordo

Al momento non è possibile fornire precise indicazioni sulle modalità di installazione del kit bike, dal momento che non è noto su quale tipo di biciclette avverrà la sperimentazione.

Si ritiene che il contenitore descritto alla sezione 3.2.2.2 potrà essere installato sulla ruota posteriore, sfruttando eventualmente l’esistenza di un portapacchi. Aver concordato un singolo contenitore con dimensioni limitate che contiene l’intero kit costituisce comunque un primo risultato importante anche dal punto di vista dell’installazione.

I dettagli della installazione del kit a bordo sono specificati nel relativo manuale [SiiM 10].

3.2.3 Funzione ATF, anti-theft feature

Questa funzione va diversificata in almeno due scenari:

- a) caso in cui il veicolo è sotto il controllo di un Utente e quindi esiste uno Smartphone associato via Bluetooth con la Piattaforma e in grado di comunicare con la centrale Sii-Mobility. Riferiamo con **IATF** (integrated anti-theft feature) questa funzione, che è realizzata sostanzialmente dal Data Harvester ed è descritta in dettaglio alla sezione 0.
- b) caso in cui il veicolo è in sosta in una rastrelliera e quindi non esiste uno Smartphone associato e le eventuali comunicazioni con la centrale Sii-Mobility vanno eventualmente realizzate in altro modo. Riferiamo con **AATF** (autonomous anti-theft feature) questa funzione. Occorre considerare due possibilità:
 - b.1 realizzare la funzione principalmente o totalmente sul veicolo, aggiungendo elementi anche complessi al kit bike ed eventuali elementi molto semplici all’infrastruttura di terra.
 - b.2 realizzare la funzione principalmente o totalmente sull’infrastruttura di terra, aggiungendo elementi anche complessi all’infrastruttura ed eventuali elementi molto semplici sul kit bike.

Dato il notevole beneficio offerto dalla funzione antifurto a utenti e gestori, in corso progetto si intende portare avanti la funzione IATF descritta in a) ed entrambi gli approcci alla funzione AATF descritti in b.1 e b.2.

In particolare, l’approccio b.2 introduce un apparato integrato con sensori, attuatori e apparati di elaborazione e comunicazione. Sarà descritto nel contesto della *Attività 3.3 - Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità* nel deliverable ([SiiM 5]).

Per quanto riguarda l’approccio b.1, la Piattaforma incorpora comunque un apparato GPS, descritto alla sezione 3.2.5.2. e un Segnale sonoro di allarme, descritto alla sezione 3.2.5.3. Se il software della Piattaforma rileva uno spostamento significativo mentre non esiste una comunicazione Bluetooth con il Data Harvester (e quindi non c’è un Utente associato) interpreta come furto questa situazione. Meccanismi diversi per rilevare l’allontanamento indebito del veicolo dalla sua posizione (per es. tag RFID) potranno essere studiati in corso lavori.

La reazione all’allontanamento indebito può essere di due tipi:

- i. viene solo azionato il Segnale sonoro di allarme;
- ii. la Piattaforma incorpora anche un apparato GSM e si mette in comunicazione con la centrale SiiM per segnalare l'evento. Questa soluzione comporta di avere stabilmente una SIM nella Piattaforma, che però viene usata solo in casi molto particolari.

Uno studio di fattibilità comparato è contenuto nella sezione 6 - *Appendice A – Studio di fattibilità della funzione AATF (autonomous anti-theft feature) realizzata nel contesto del kit bike.*

Dai risultati dello studio, la soluzione ii. non è perseguibile nei tempi e costi di progetto e quindi il Kit includerà un apparato GPS ma non un apparato GSM.

3.2.4 Smartphone utente

3.2.4.1 Diagramma di contesto

La *Figura 11* mostra il diagramma di contesto dello Smartphone Utente

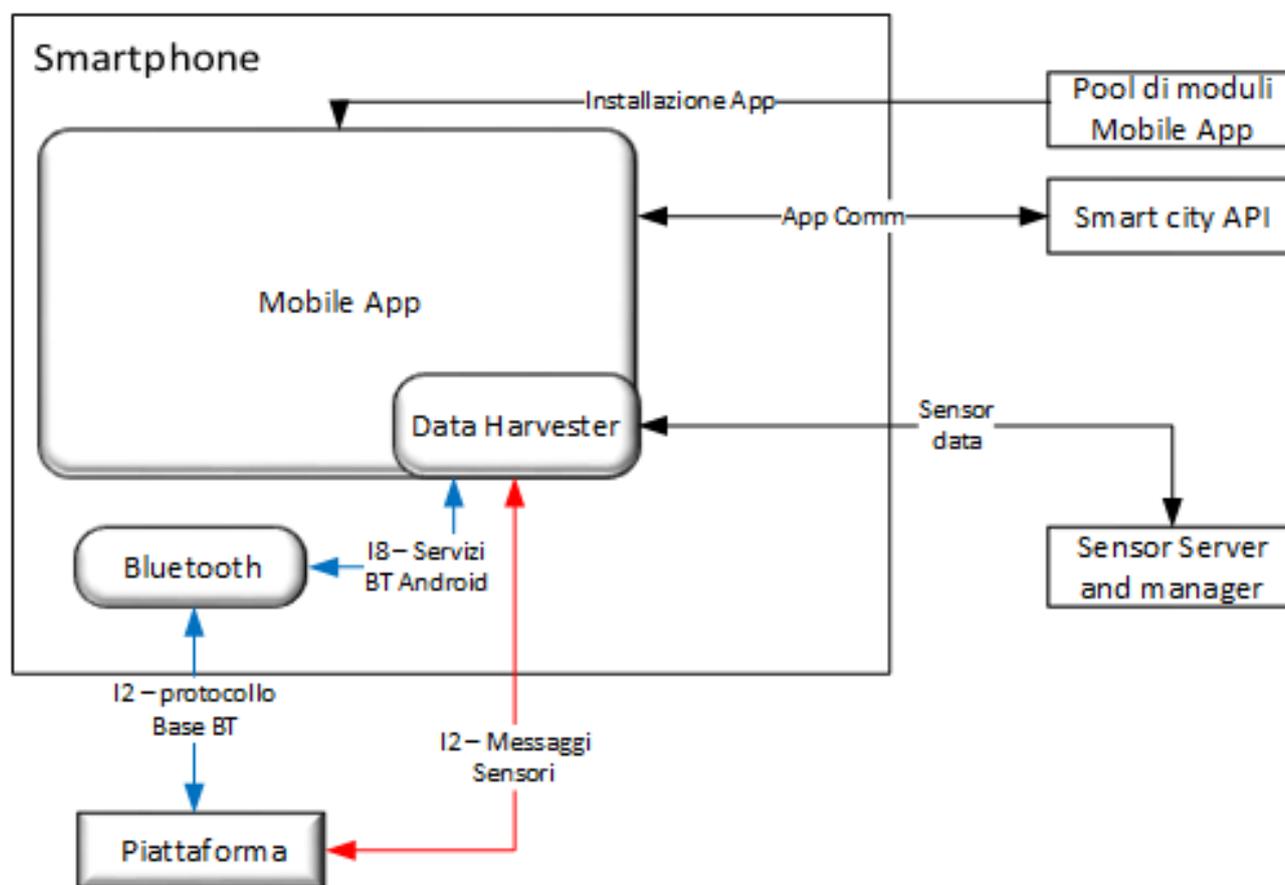


Figura 11 : Smartphone – diagramma di contesto

Di seguito sono indicate le caratteristiche minime che lo Smartphone dell'Utente deve possedere per supportare efficacemente gli elementi del Kit che vi sono installati:

- ✚ Sistema Operativo **Android 4.2.1 Jelly Bean** o revisioni successive retro-compatibili;
- ✚ Bluetooth Low Energy
- ✚ GPS

3.2.4.2 Data Harvester, MAPP16

Il Data Harvester (MAPP16) nelle sue varie istanze e versioni è descritto in altri documenti che al momento non sono determinati.

3.2.4.3 MobileAppBike, APP01

La Mobile App APP01 è descritta in altro documento di progetto, che al momento non è determinato.

3.2.4.4 IATF, integrated anti-theft feature

L'allontanamento del veicolo dall'Utente viene rilevato per assenza, per un periodo configurabile, dello scambio periodico di messaggi di vitalità tra Piattaforma e Data Harvester.

Questa situazione viene interpretata come possibile furto dal Data Harvester, che informa l'Utente, autonomamente o coinvolgendo la Mobile App. L'Utente può resettare la segnalazione nel caso in cui si è allontanato volontariamente dal veicolo, oppure confermare, nel qual caso la centrale Sii-Mobility viene informata per le azioni del caso.

3.2.5 Sensori / Attuatori

Ad eccezione dei Sensori ambientali, sviluppati nel contesto dell'Attività 3.2 e descritti in [SiiM 4], tutti i sensori e attuatori collegati con la Piattaforma del kit bike sono sviluppati appositamente dall'Attività 3.1 e descritti nelle sezioni seguenti fino al livello di dettaglio.

3.2.5.1 Piattaforma inerziale

Il rilevamento delle sollecitazioni meccaniche del veicolo si concretizza nell'invio tramite Bluetooth al Data Harvester e quindi alla centrale SiiM di un set di dati, eventualmente includendo valutazioni preliminari sullo stato della strada e sulla condotta del veicolo. Le valutazioni finali saranno eseguite sulla centrale SiiM, su base statistica (molti veicoli che passano sullo stesso tratto di strada) e correlando la localizzazione dei dati (fornita con la precisione dello Smartphone Utente) in modo che il comportamento scorretto di un singolo Utente (es. bicicletta che sale e scende dai marciapiedi) possa essere riconosciuto come dato spurio: la Piattaforma, il Data Harvester e la Mobile App non sono in grado di svolgere queste elaborazioni.

Il sistema di monitoraggio stradale si poggia prevalentemente su misure effettuate mediante la Piattaforma Inerziale embedded nella motherboard descritta alla sezione 3.2.2.3.3

3.2.5.2 Sensore GPS antifurto

La funzione Autonomous Anti-Theft Feature (AATF), Sezione 3.2.3 scenario b.1.i., si concretizza con il tracking della posizione GPS, (caso d'uso UC6 alla Sezione 3.2.2.4, stato Idle.PreAlert dello State chart. In questo scenario la Piattaforma non è associata con uno Smartphone quindi non è in grado di sfruttare il GPS di quest'ultimo. Analogamente non è possibile sfruttare la connettività dello Smartphone. Per queste ragioni è necessario che la Piattaforma disponga di una propria scheda GPS per poter monitorare lo spostamento del mezzo.

La funzione AATF si può dettagliare nel modo seguente:

1. AATF con segnale sonoro di allarme, Sezione 3.2.4.3, senza modulo di comunicazione long range. Nel caso in cui la Piattaforma si trova in un'area di sosta la funzionalità AATF può essere coadiuvata dai tag RFID della Piattaforma e della Rastrelliera Intelligente.

Sul mercato è disponibile un'ampia gamma di soluzioni hardware di schede GPS. Le caratteristiche della scheda GPS devono garantire prestazioni sufficientemente adeguate all'implementazione della

funzionalità AATF e contemporaneamente sottostare al vincolo del contenimento dei costi, nell'ottica di una produzione in serie della Piattaforma e per facilitarne la diffusione. In accordo con le caratteristiche generali appena esposte è stato scelto il modulo denominato VK2828U7G5LF V1.0 dotato di una antenna GPS di 25 x 25 x 4 mm con supporto al A-GPS (Figura 12).

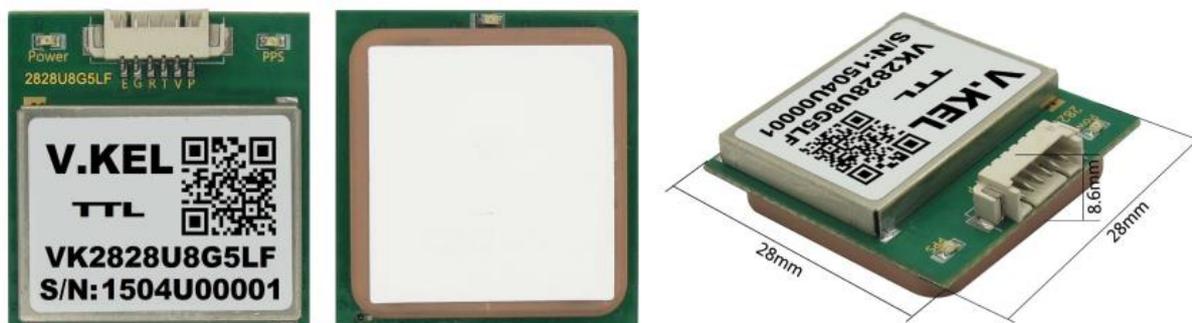


Figura 12: modulo GPS VK2828U7G5LF V1.0

3.2.5.2.1 Interfaccia fisica

L'installazione della scheda GPS con la board Arduino si concretizza attraverso i collegamenti fisici. L'interfaccia fisica è costituita da 6 collegamenti descritti nella tabella seguente.

Nome	Descrizione
PPS	Pulse Per Second, indica l'inizio di un secondo
Vcc	Alimentazione
TX	Interfaccia UART/TTL
RX	Interfaccia UART/TTL
GND	Ground
EN	Indica se il dispositivo è acceso

Tabella 3: Interfaccia fisica del modulo GPS

La scheda GPS opera a valori di tensione compresi tra +3.3V e +5V e il consumo medio è di 50mAh. La temperatura operativa supportata è compresa tra -40 °C e +85 °C.

3.2.5.2.2 Interfaccia logica

La scheda GPS è collegata tramite interfaccia Transistor-Transistor Logic (TTL). La configurazione seriale è la seguente:

- Data bits: 8 bit
- Parity: None
- Stop bits: 1
- Handshake: None

Il dato fornito dalla scheda GPS è conforme allo standard di comunicazione dati NMEA 0183. La struttura generale del messaggio è del tipo:

\$PREFISSO, dato1, dato2 ... datoN-1, datoN * CHECKSUM

dove il simbolo \$ rappresenta l'inizio del messaggio e i caratteri di terminazione sono CR LF. I messaggi utilizzano la codifica di caratteri ASCII tra 0x20 (space) e 0x70 (~) e la lunghezza massima è di 80 simboli. I caratteri speciali sono riportati in tabella

ASCII	Hex	Decimale	Descrizione
<CR>	0x0d	13	Ritorno a capo
<LF>	0x0a	10	Nuova riga o a capo
\$	0x24	36	Inizio
*	0x2a	42	Delimitatore checksum
,	0x2c	44	Delimitatore di campo
!	0x21	33	Inizio di incapsulamento
\	0x5c	92	TAG block
^	0x5e	94	Delimitatore per HEX
~	0x7e	126	Riservato

Tabella 4: Interfaccia logica del modulo GPS

Una stringa è composta dai seguenti campi:

- GGA: time, position, position type
- GLL: latitude, longitude, UTC
- GSA: GPS receiver operating mod, satellites for positioning, DOP value
- GSV: Available GPS satellites information, azimuth, elevation, SNR
- RMC: time, date, position, speed
- VTG: the speed information on ground
- MSS: signal strength

Esempio:

\$GPGGA,060556.00,2236.91418,N,11403.24669,E,2,08,1.02,115.1,M,-2.4,M,,0000*43

Nome	Esempio	Unità di misura	Descrizione
ID	\$GPGGA		Head protocollo GGA
UTC	060556.00		hhmmss.ss

Latitudine	2236.91418		ddmm.mmmmm
Direzione N/S	N		N=nord, S=sud
Longitudine	11403.24669		dddmm.mmmmm
Direzione E/W	E		W=ovest, E=est
Direzione posizione	2		0-3
#satelliti	8		0-12
HDOP	1.02		Accuratezza orizzontale
MSL	115.2	Metri	
Unità di misura	M	Metri	
Ground	-2.4	Metri	
Unità di misura	M	Metri	

Tabella 5: esempio di messaggio NMEA 0183

3.2.5.2.3 Architettura

L'architettura di riferimento è schematizzata nella Figura seguente. La scheda GPS si interfaccia con la Piattaforma attraverso l'interfaccia I1 – Messaggi/Seriali. La Piattaforma ha il compito di acquisire il segnale ed elaborarlo al fine di implementare le funzionalità richieste.

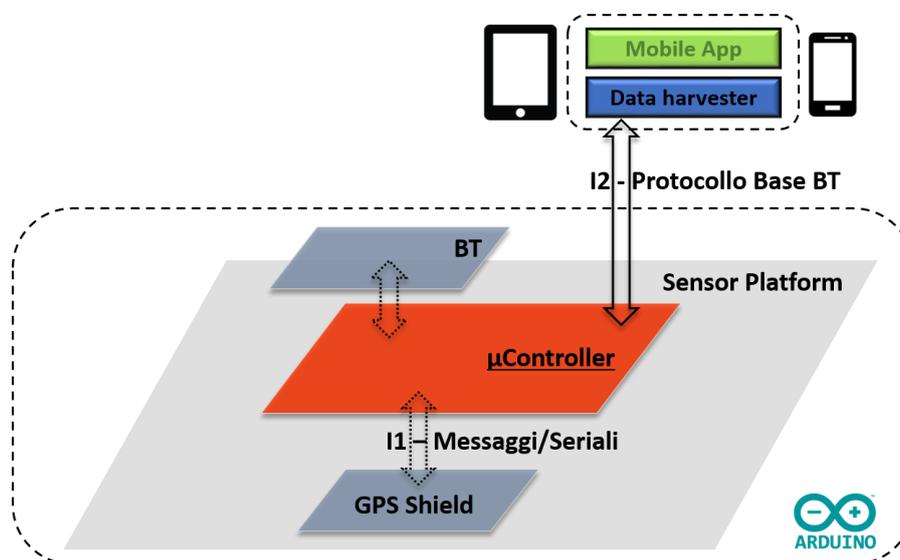


Figura 13: Scheda GPS nel contesto della architettura del Kit bike

3.2.5.2.4 Specifica di dettaglio

Di seguito sono riportati i parametri tecnici della scheda GPS VK2828U7G5LF

Proprietà del modulo

Chip	UBX-M8030-KT
C/A (Data Rate)	1.023 MHz
Frequenza di ricezione	L 1 [1575.42 MHz]
Canale di ricezione	56
Proprietà di posizionamento	
Velocità	< 0.1 m/s
Altezza massima	50 km
Velocità massima	500 m/s
Accelerazione	< 4G
Altre proprietà	
Start	1 [s] in media
A-GPS	3 [s]
Data baud	9600 (default) altre: 1200, 2400, 4800, 19200, 38400, 57600, 115200, 230400, 460800, 921600
Temperatura operativa	da -40°C a 85°C

Nota: emerge che il segnale GPS ha un proprio orario di riferimento. Da valutare se è possibile ottenere il clock direttamente dalla Shield GPS evitando la fase di sincronizzazione descritta nella Sezione 3.2.5.2, rappresentata dal caso d'uso UC5 e dallo stato Associated.Synchronization.

3.2.5.3 Attuatore Segnale sonoro di allarme

Al momento si prevede di installare una piccola sirena che funzioni da antifurto nel momento in cui il sistema Autonomous Anti Theft rivela uno spostamento anomalo. La sirena suonerà nel rispetto della normativa sugli antifurti di veicoli.



La sirena è alimentata dalla linea a 12V, e comandata tramite relè da una porta digitale della Piattaforma Arduino, con livello sonoro di 105dBA, e a una frequenza di 1.5kHz, con dimensioni 51 x 51 x 35 mm. Il relè è installato sulla proto-shield aggiuntiva descritta alla sezione 3.2.2.3.4.

Ci si riserva di indagare in una seconda fase l'opportunità di usare un ulteriore segnalatore acustico a bordo veicolo ad uso del sensore di prossimità descritto nel paragrafo Sensori anti-collisione.

Quando il sensore di prossimità infatti rileva un oggetto vicino / in avvicinamento manda un segnale che viene riportato al Data Harvester (MAPP16) e fa scattare un segnale sonoro sullo Smartphone utente.

Nel caso si ritenesse opportuno segnalare ulteriormente l'eventuale ostacolo attraverso un segnalatore acustico sulla Piattaforma, si può prevedere l'uso di un buzzer, con opportuno amplificatore da alloggiare sulla proto-shield aggiuntiva, comandato da un piede PWM dell'Arduino, di cui è possibile modulare la frequenza, oltre alla durata del suono.

3.2.5.3.1 Interfaccia fisica e logica

È costituita da una singola porta di I/O di Arduino 101, identificata in Tabella 2

3.2.5.3.2 Architettura

Data la semplicità dell'apparato non occorre dettagliare oltre la sua descrizione.

3.2.5.4 Sensori ambientali

Il sistema sarà sviluppato in modo da essere compatibile con la rilevazione delle seguenti grandezze:

- NO₂
- NO
- O₃
- H₂S
- SO₂
- CO
- VOC (Volatile Organic Compound)

Che rappresentano i principali inquinanti anche se nel Kit Bike verrà implementato un subset di tali sensori, in particolare si utilizzerà ([SiiM 8]) il modulo AFE per 2 sensori (NO₂ e H₂S) per limitare gli ingombri, i consumi e il carico della motherboard.

I sensori ambientali per il Kit bike sono descritti nel documento [SiiM 4]. In particolare:

- l'interfaccia fisica è descritta alla sezione 3.1.1.2 e 3.2.3.2.
- l'interfaccia logica è descritta alla sezione 3.1.1.3 e 3.2.3.3

I sensori dovranno essere implementati seguendo le prescrizioni di interfacciamento meccanico descritte nel paragrafo 3.2.2.2 del documento.

3.2.5.5 Sensori anti-collisione

Nella scelta del sensore di anticollisione sono stati definiti i requisiti fondamentali:

- Costo ridotto in vista di una produzione in serie del kit;
- Ingombro ridotto nell'ottica della realizzazione finale di un prototipo;
- Distanza massima rilevabile dell'ordine di qualche metro: sensori con distanza massima rilevata al di sotto del m non sono considerati in quanto inefficienti per l'applicazione;
- Compatibilità con piattaforma Arduino (livello della tensione di alimentazione, corrente assorbita, protocollo di comunicazione).

Sono state analizzate tre differenti tecnologie che si differenziano soprattutto per performance, dimensioni e costo del sensore:

- Laser (misura della distanza in funzione del tempo di volo, triangolazione o comparazione di fase);
- IR (misura della distanza tramite triangolazione);
- Ultrasuoni (misura della distanza in funzione del tempo di volo del fascio ultrasonico).

La tecnologia scelta è quella ad ultrasuoni in quanto soddisfa sia il requisito di performance (distanza massima dell'ordine dei 4 m) sia requisiti di costo e dimensioni ridotte. Il modello utilizzato è il sensore SRF05, che è una evoluzione del sensore HC SR04 utilizzato in una prima fase di sperimentazione. Il sensore scelto è in grado di raggiungere portate di 450 cm ed è compatibile con la logica operativa di 3.3 V di Arduino 101 (piattaforma): prove sperimentali hanno messo in luce come con Arduino 101 la portata del sensore HC SR04 fosse limitata a circa 120 cm mentre per distanze maggiori è stata riscontrata una pesante instabilità della misura.

Le caratteristiche principali del sensore sono indicate nella Tabella 6.

PARAMETRI	HY SRF05
Tensione di Lavoro	5 Vdc
Corrente Assorbita	< 2mA
Segnale di Uscita	Livello alto /basso: (5 /0) V
Angolo di Inclinazione	< 15 °
Range Operativo	2-450 cm
Precisione	~2 mm
Input trigger Signal	Impulso 10 µs TTL
Echo Signal	Uscita TTL
Pins:	<ul style="list-style-type: none"> • VCC • trig(T) • echo(R) • OUT • GND

Tabella 6: caratteristiche principali del sensore HY-SRF05

Tabella 6 mette in luce la presenza di un piedino OUT che serve per impostare la modalità operativa del sensore: mettendo il pin a massa il sensore funziona a singolo pin che in fasi differenti e successive opera come trigger e come echo, mentre se tale pin non viene collegato il funzionamento è a 2 pin distinti. Il sensore ha un resistore di pull-up interno in modo tale da gestire la condizione operativa (che è quella adottata) di funzionamento a 2 pin (in linea con il funzionamento del sensore HC SR04). Gli aspetti da considerare sono la massima distanza rilevabile (vedi Figura 14a) e il posizionamento del sensore (vedi Figura 14b). Il sensore è stato testato sperimentalmente verificando come, in condizioni di superficie indagata estesa rispetto la larghezza del fascio, la distanza massima misurabile sia di circa 430 cm con una inclinazione massima di 25°, oltre la quale le performance del sensore non sono più affidabili.

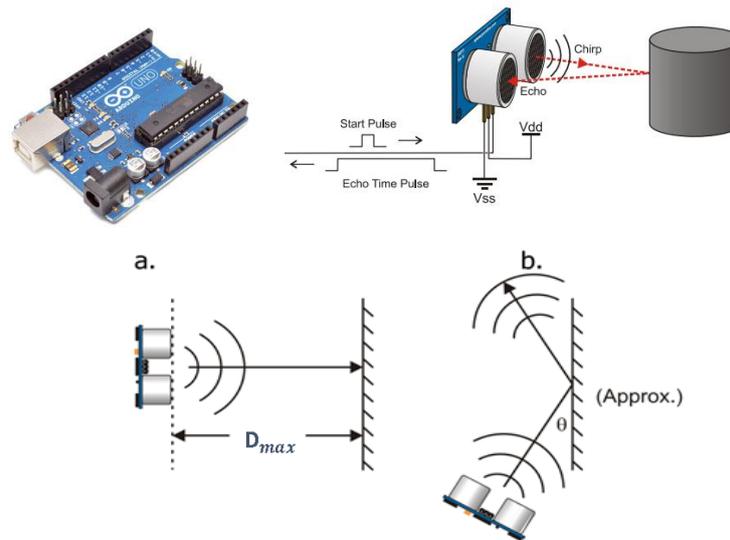


Figura 14: Principio di funzionamento del sensore ultrasonico con particolare attenzione verso il posizionamento del sensore

Il principio di funzionamento del sensore si basa sul tempo di volo dell'onda trasmessa a seguito di una eccitazione (*StartPulse*, vedi Figura 27): tale valore dipende dalla velocità di propagazione dell'onda la quale è afflitta da un contributo dato dalla temperatura, come segue:

$$V = (331.4 + 0.62 T) \text{ m/s}$$

dove T è la temperatura e V è la velocità di propagazione dell'onda ultrasonica.

Per tale motivo, in aggiunta al sensore ultrasonico, è stato scelto anche un sensore di temperatura in modo tale da ottimizzare la misura della distanza. Contrariamente a quanto sostenuto nella versione a del documento, il sensore di temperatura DHT-11 non è stato adottato nella soluzione attuale ma è stato preferito il modulo DS18B20 di facile interfacciamento con la piattaforma.

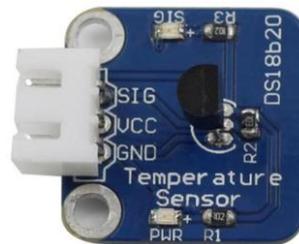


Figura 15: modulo sensore di temperatura DS18B20

Le principali caratteristiche sono:

- Comunicazione tramite 1-Wire bus;
- Temperatura direttamente in ° Celsius;
- Range di Temperatura rilevabile: -55°C ~ +125°C, errore di 1°C.
- Resistenza a disturbi;
- Tensione Operativa: 3.3-5V DC;
- PCB size: 2.0 x 2.0 cm;
- Sensore digitale con risoluzione di 12 bit.

Il modulo presenta internamente una resistenza di pull-up, vedi Figura 29.

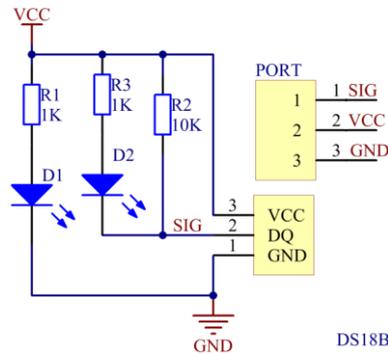


Figura 16: resistenza di pull-up interna al modulo DS18B20

3.2.5.5.1 Interfaccia fisica

La descrizione delle interfacce fisiche dipende dalla definizione pin di In/Out del sensore ultrasonico e di temperatura (vedi Tabella 2 in 3.2.2.3.4).

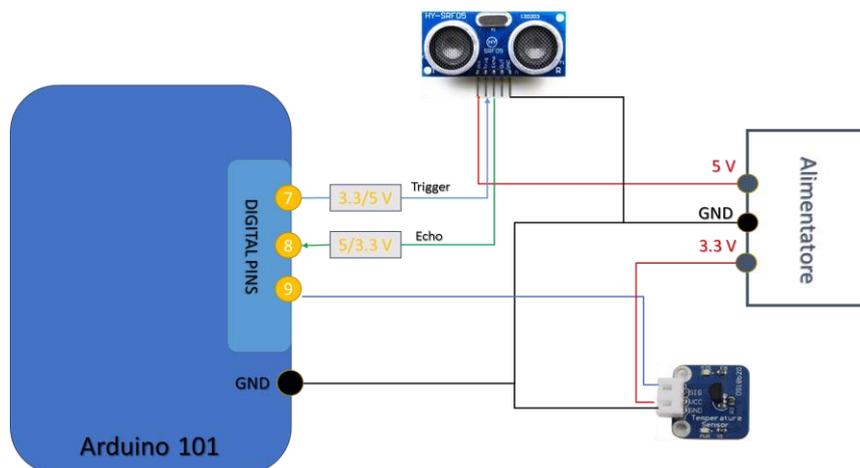


Figura 30: interfaccia fisica tra Arduino101 sensore ultrasonico HY SRF05 e modulo sensore di temperatura DS18B20

Figura 30 illustra i collegamenti del sensore ultrasonico HY SRF05 e del sensore di temperatura: il primo lavora a 5 V mentre il secondo ad una tensione di 3.3 V. I collegamenti del sensore sono inerenti alla sua alimentazione e gestione del fascio ultrasonico, mentre il sensore in temperatura ha una logica 1-Wire bus.

3.2.5.5.2 Interfaccia logica

Il sensore inizialmente scelto HC SR04 è stato sostituito dal HY SRF05 in quanto le sue prestazioni non sono state considerate insoddisfacenti: è stato possibile stabilizzare le rilevazioni fino a 120 cm attraverso filtri che avrebbero d'altro canto limitato e vanificato le performance dello scheduler di gestione dei processi. Per scopi cautelativi è stato adottato un traslatore di livello logico 5/3.3 V per gestire l'interfacciamento tra sensore e piattaforma (vedi Figura 30, 31).

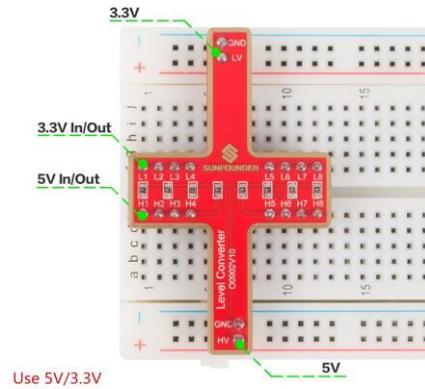


Figura 17: Traslatore di livello logico

I segnali logici scambiati sono di eccitazione del sensore (attraverso un input logico di $10\ \mu\text{S}$), a cui fa seguito un burst ultrasonico che porta alla ricezione dell'echo riflesso dal target, logicamente corrispondente ad un segnale TTL positivo, di durata proporzionale alla distanza rilevata (vedi Figura 32).

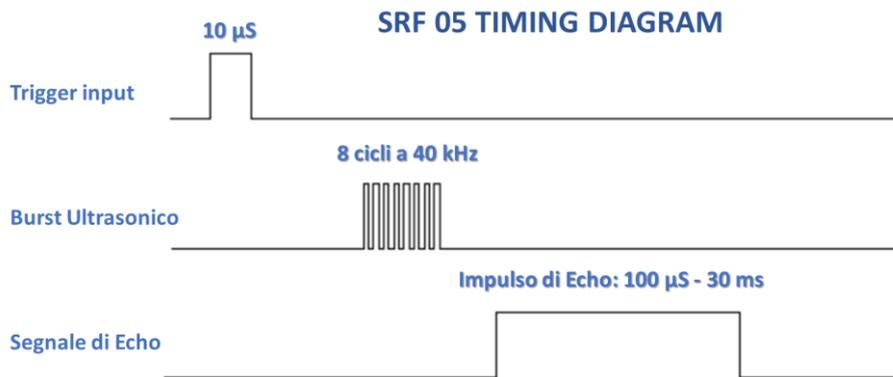


Figura 32: Diagramma temporale del funzionamento del sensore

3.2.5.5.3 Architettura

L'architettura generale all'interno della quale si colloca il sensore di prossimità è descritta in Figura 33: in particolare il sensore (RangeFinder) dovrà interfacciarsi con la piattaforma la quale ha a carico l'acquisizione del segnale, l'eventuale adattamento del segnale ai canali ADC di ingresso e la trasmissione dell'informazione della distanza rispetto il veicolo che segue attraverso comunicazione Bluetooth verso Smartphone usufruendo dello Shield Bluetooth di cui sarà dotata la piattaforma. L'allarme viene abilitato a distanza ridotta tra bike e veicolo che segue e viene gestito come notifica da un applicativo su Smartphone che deciderà la logica di intervento. L'architettura del kit anticollisione fa riferimento a quanto indicato in Figura 30.

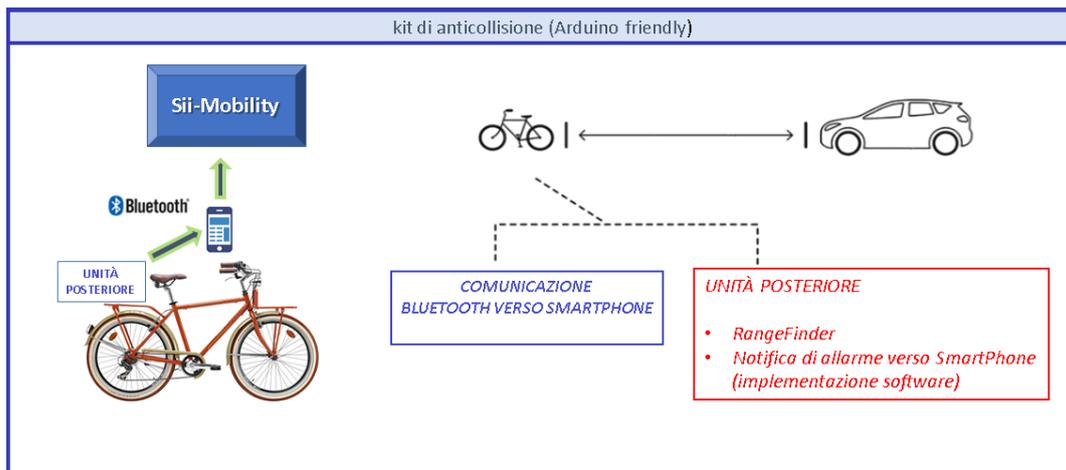


Figura 18: Logica dell'architettura del kit anti collisione

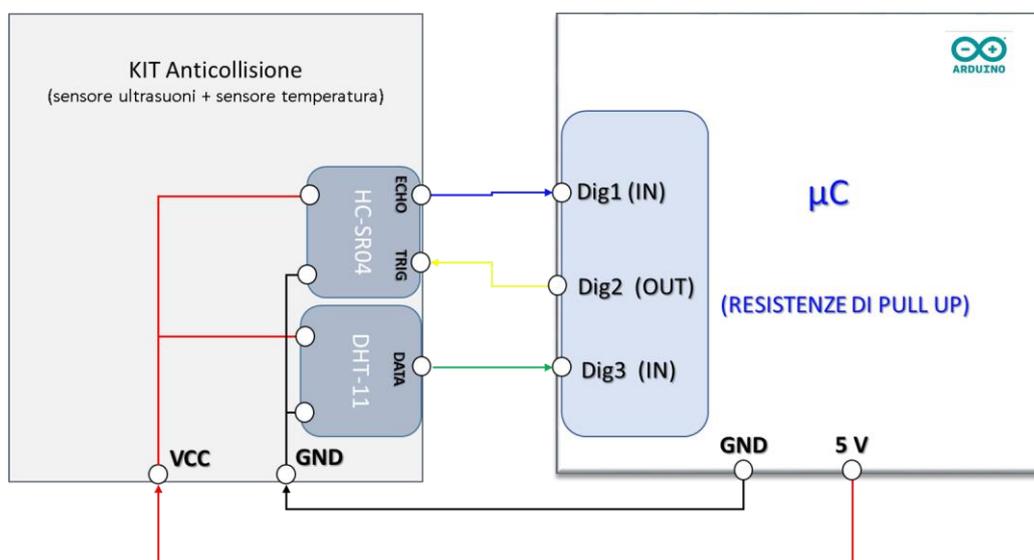


Figura 19: Architettura del sistema (KIT anticollisione)

3.2.5.5.4 Specifica di dettaglio

Il sensore ultrasonico HY SRF05 ha caratteristiche del pattern di radiazione del tutto simili al sensore HC SR04 utilizzato in prima fase: sono riportati di seguito tre tipologie di sensori ad ultrasuoni che si differiscono per larghezza del fascio ultrasonico e costo (vedi Tabella 7, Figura 34).

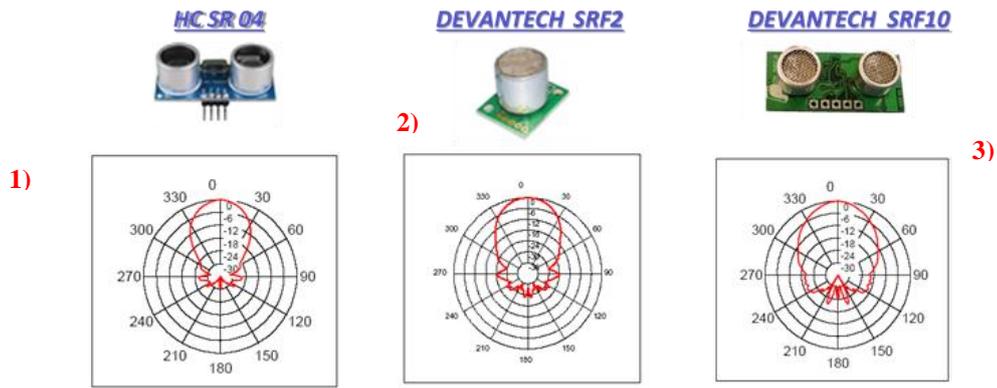


Figura 20: tre sensori acustici scelti come possibile elemento sensibile ai fini del rilevamento della prossimità tra bike e veicolo che segue

La prima soluzione comporta un fascio che offre un'attenuazione di 12 dB per un angolo di 33° rispetto la longitudinale, mentre la terza soluzione è dotata di un fascio molto largo con un'attenuazione di 12 dB a 58°.

La Tabella 6 mette in luce le principali caratteristiche dei sensori sopraelencati, in termini di tensione di alimentazione, corrente assorbita, distanza massima rilevata, costo e dimensioni del sensore. Le specifiche del sensore HY SRF05 sono del tutto simili al HC-SR04 con il vantaggio che non è stata riscontrata una diminuzione della portata del sensore se acquisito da board Arduino 101.

PARAMETERO	HC SR04	DEVANTECH SR02	DEVANTECH SRF10
V_{ALIM} [V]	5	5	5
I_A [mA]	15	4	15
R_{MAX} [cm]	400	600	1100 (programmabile)
Costo [€]	0.80	9	30
Dimensioni [mm]	45 x 20 x 15	24 x 20 x 17	43 x 20 x 17
Interfaccia	TTL	IIC/TTL	IIC
Frequenza di lavoro [kHz]	40	40	40

Tabella 6: Caratteristiche principali dei sensori di prossimità ultrasonici esaminati per il kit di anticollisione

Per la selezione del sensore è stata fatta anche un'analisi delle piattaforme Arduino presenti sul mercato, al fine di individuare le interfacce messe a disposizione per le comunicazioni con dispositivi esterni. In particolare:

PARAMETRO	ARDUINO UNO (rev. 3)	ARDUINO MEGA 2560	ARDUINO LEONARDO	ARDUINO DUE	ARDUINO MICRO	GENUINO 101
MCU	ATmega328	ATmega2560	ATmega32U4	AT SAM3X8E	AT32U4	INTEL CURIE
Clock (MHz)	16	16	16	84	16	32
FLASH (KB)	32	256	32	512	32	196
SRAM (KB)	2	8	2.5	96	2.5	24
EEPROM (KB)	1	4	1	-	1	-
I/O	20 (14 dig, 6 analog)	70 (54 dig, 16 analog)	20 (8 dig, 12 analog)	66 (54 dig, 12 analog)	20 (8 dig, 12 analog)	20 (14 dig, 6 analog)
PWM (pin)	6	15	7	12	12	12
ADC	6 canali a 10 bit	16 canali a 10 bit	16 canali a 10 bit	12 canali a 10 bit	12 canali a 10 bit	12 canali a 10 bit
V _{op} (V)	5 / 3.3	5	5	3.3	3.3	3.3 (5 V tolerant I/O)
Comunicazione	1 Seriale, 1 I2C, 1 SPI	4 Seriale, 1 I2C, 1 SPI	4 Seriale, 1 I2C, 1 SPI	4 Seriale, 2 I2C, 2 SPI, 1 CAN	1 Seriale, 1 I2C, 1 SPI	1 Seriale, 1 I2C, 1 SPI
Peculiarità	Molti librerie e shield presenti	Molte linee di I/O	Comunicazione diretta PC con USB	Elevata potenza di calcolo	Dimensioni ridotte	Modulo BT, accelerometro/giros copio triassiale

Tabella 7: Caratteristiche principali e peculiari di alcuni dei prodotti Arduino/Genuino presenti sul mercato

Le schede analizzate in Tabella 8 mettono a disposizione protocolli di comunicazione (IIC/SPI/TTL) compatibili con le specifiche dei sensori analizzati in Figura 34. Una volta selezionata la piattaforma, Arduino 101, la scelta finale è ricaduta sul sensore HY SRF05 come migliore compromesso tra costi e performance.

La fase di sviluppo del sistema comporterà delle valutazioni in termini di:

- Scelta di installare o meno una schiera di sensori per avere un fascio di copertura maggiore;
- Nel caso di installazione di una schiera di elementi sensibili scelta del modo di attivazione di questi, che potrebbe avvenire in serie per evitare interferenze.
- Disposizione di ogni singolo sensore in modo tale da evitare ritorni spuri.
- Progettazione di uno chassis sia per il sostegno del sensore che per evitare ritorni spuri sul pin Echo del sensore.

In una prima fase di sperimentazione è stato osservato come la lettura del sensore diventi non affidabile con un angolo di inclinazione superiore a circa 25/30°: ci si riserva la possibilità di acquistare in un secondo momento un sensore Devantech SRF10 qualora il suo costo e performance siano ritenute rispettivamente accettabili e necessarie per le finalità del progetto.

3.2.6 Protocollo base Bluetooth

In questa sezione si sviluppa la specifica del protocollo di base Bluetooth, che viene indicato come *I2 – protocollo Base BT* in *Figura 6* e *Figura 11*.

Tale specifica deve essere condivisa tra chi sviluppa il software della Piattaforma e del Data Harvester, in modo da utilizzare in modo consistente le funzionalità Bluetooth presenti sulla Piattaforma e sul Tablet per realizzare un meccanismo di trasporto wireless di messaggi.

3.2.6.1 Protocollo di scambio dati

Ci si propone di sviluppare il protocollo seguendo lo schema di comunicazione “nativo” Bluetooth Low Energy, descritto in 3.2.2.3.2. Qualora in corso lavori il throughput si rivelasse insufficiente per supportare tutte le funzioni previste, si provvederà ad un riesame generale del meccanismo per migliorarlo o per selezionarne uno alternativo.

3.2.6.2 Sincronizzazione del clock di Piattaforma

In caso di temporanea perdita di connessione Bluetooth tra Piattaforma e Smartphone utente, i dati dei sensori devono essere memorizzati nella Piattaforma per essere poi inviati al Data Harvester appena possibile. Di conseguenza devono essere geo referenziati e datati all’origine dalla Piattaforma. Questo significa che la Piattaforma deve includere un apparato GPS e che il real-time clock della Piattaforma deve essere allineato a quello dello Smartphone.

Il clock di una scheda come Arduino ha tipicamente una deriva di 0.5 secondi ogni ora. Questo produce uno scarto di 12 secondi ogni giorno. Pertanto per una corretta gestione dei dati da parte del Data Harvester, Mobile App e Sii-Mobility si rende necessario la definizione di un ulteriore caso d’uso che ha come scopo la sincronizzazione del clock della scheda Arduino. La soluzione è un algoritmo di sincronizzazione che si articola nei seguenti punti:

1. La Piattaforma invia una richiesta di sincronizzazione allo Smartphone utente;
2. Lo Smartphone utente risponde con un messaggio contenente la data misurata con il proprio orologio;
3. La Piattaforma aggiorna il clock locale con il valore opportuno.

4 Kit carbus base

Il *Kit carbus base* è progettato per equipaggiare autobus di proprietà di gestori di TPL. Costituisce una versione con prestazioni ridotte, ma comunque rilevanti, rispetto al Kit carbus evoluto descritto nel seguito.

È sviluppato per poter equipaggiare un numero rilevante di autobus con costi contenuti, questo grazie al fatto che il partner ELFI concede in comodato d'uso al Progetto un certo numero di dispositivi da installare a bordo ed inoltre la possibilità di utilizzare una centrale operativa proprietaria che li comanda e controlla e raccoglie dati di infomobilità.

4.1 Architettura generale

L'architettura generale, contestualizzata in Sii-Mobility, del Kit è mostrata in *Figura 21*.

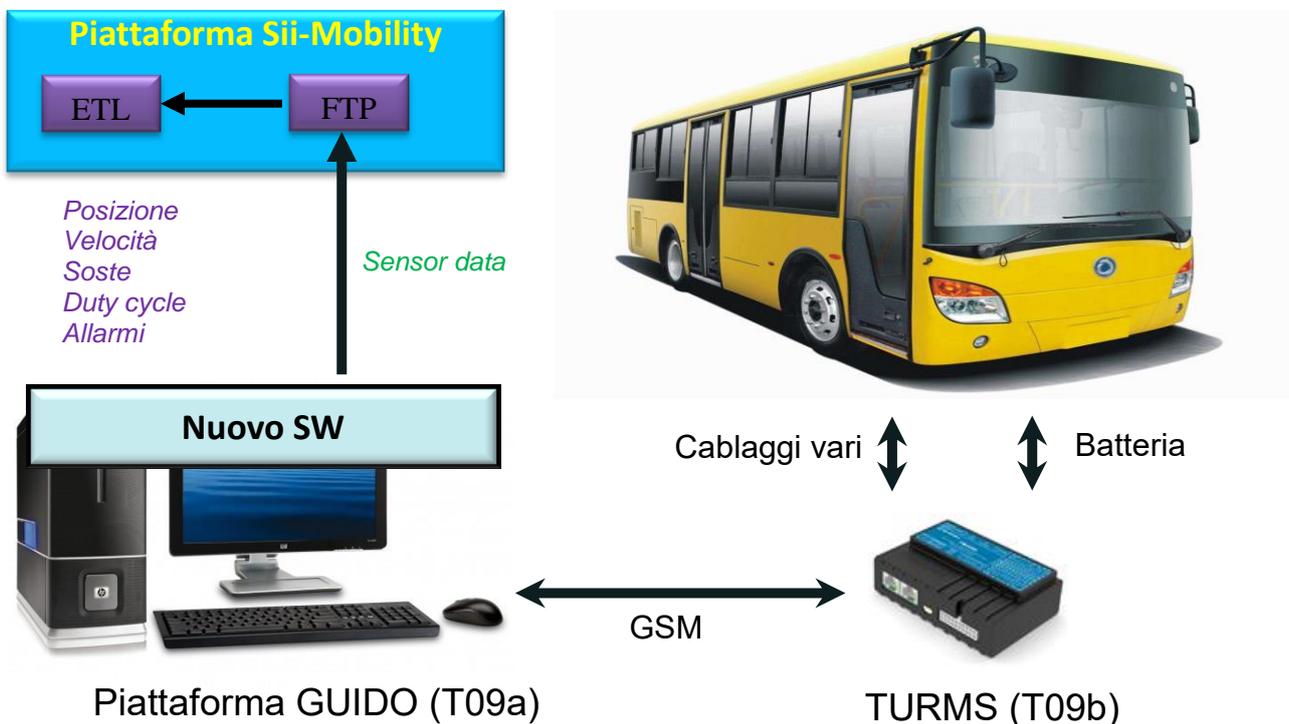


Figura 21 : Architettura generale del Kit carbus base

Le principali componenti del sistema già esistente (che costituisce un Tool nella terminologia utilizzata in [SiiM 2]) sono:

- **TURMS** è un apparato da installare a bordo autobus. Non ha una interfaccia verso il conducente, può essere localizzato in posizione nascosta nel cruscotto. È provvisto di omologazione;
- **Piattaforma GUIDO** è una centrale operativa che raccoglie dati da un numero di apparati TURMS e presenta i dati dei singoli veicoli e quelli cumulativi di una flotta i veicoli.

Questi elementi vengono forniti in comodato d'uso gratuito dal partner ELFI (un numero concordato di apparati TURMS e un numero corrispondente di licenze sulla Piattaforma GUIDO) e pertanto non costituiscono costi rendicontabili per il Progetto Sii-Mobility. Non verranno descritti in maggior dettaglio nel seguito.

Costituiscono invece costi rendicontabili le attività di:

- **modifica** del Software esistente della Piattaforma GUIDO per l'invio dei dati tramite protocollo FTP su ftp.disit.org;
- **installazione** degli apparati TURMS sugli autobus per la sperimentazione;
- **acquisto** di eventuali apparati TURMS e licenze GUIDO oltre il numero concordato.

4.2 Specifica di dettaglio

4.2.1 Diagramma di contesto

La *Figura 22* mostra un primo livello di specifica e ripropone lo schema generale di architettura nella notazione utilizzata per la descrizione di dettaglio. A questo livello occorre definire con maggior precisione le interfacce mostrate.

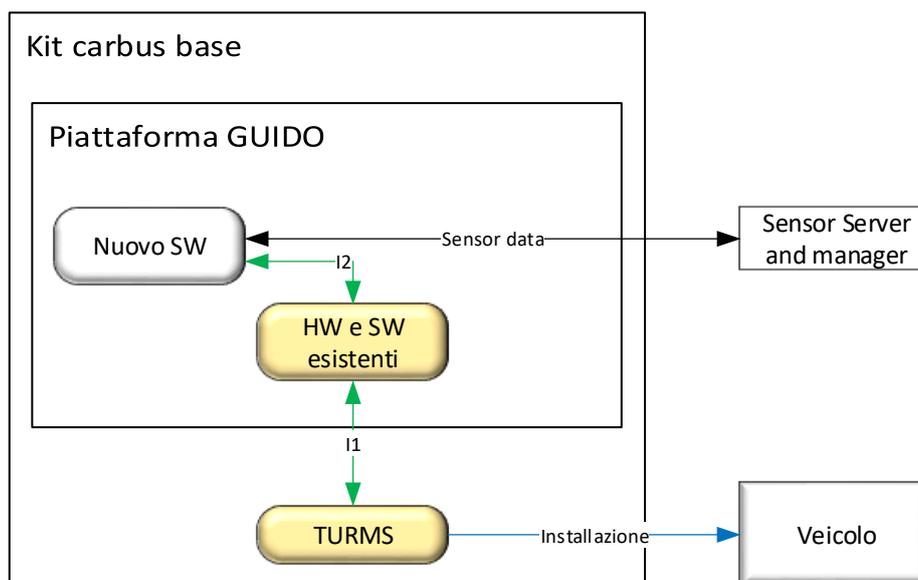


Figura 22 : Kit carbus base - diagramma di contesto

In particolare:

- Interfacce esterne:
 - *Sensor data* è una interfaccia applicativa realizzata con scambi di dati via FTP. Viene dettagliata alla sezione 4.2.2;
 - *Installazione* dell'apparato TURMS è un'interfaccia fisica che coinvolge lo studio delle modalità di interfacciamento con gli apparati dell'autobus. L'argomento è sviluppato alla sezione 4.2.4;
- Interfacce interne:
 - **I1** è l'interfaccia fisica e logica tra Piattaforma GUIDO e apparati TURMS. Non è rilevante per Sii-Mobility e non viene ulteriormente sviluppata in questo documento;
 - **I2** rappresenta una interfaccia interna per ELFI e non viene ulteriormente sviluppata.

4.2.2 Interfaccia GUIDO – Sii-Mobility

Il sistema Guido è un gestionale di controllo avanzato per la gestione, controllo ed ottimizzazione delle flotte di autoveicoli basato su dispositivi di geo referenziazione e comunicazione tempo reale dei dati via GPRS/UMTS.

L'interfaccia di controllo si basa su Web 2.0 dove l'utente interagisce con un sistema efficace e dinamico basato sul dinamismo grafico delle mappe e della interazione rapida con le funzionalità di cui dispone "Guido".

L'interfaccia di controllo usa i dati di posizione e velocità che i dispositivi TURMS inviano e che sono ospitati presso database dedicato, che viene opportunamente interrogato dal sistema software per permettere la visualizzazione in tempo reale.

Il sistema è basato su un database relazionale MySQL versione 5.x.

I dati inviati dai vari dispositivi GPS sono inviati in codifica NMEA verso il servizio server che scompatta i campi NMEA e popola i campi corrispondenti sulla specifica tabella del database.

I dati sono quindi organizzati secondo l'identificativo del dispositivo, la data, l'ora, la posizione, la velocità ed informazioni di stato dispositivo.

Al fine di non interferire con i dati degli altri utenti che hanno installato TURMS ma non fanno parte del progetto Sii-Mobility, si prevede un accesso al database con cui recuperare i dati inerenti ai solo dispositivi registrati come dominio Sii-Mobility.

L'interfaccia tra sistema GUIDO e sistema Sii-Mobility prevede una scrittura periodica dei dati contenuti in GUIDO in un server FTP del sistema (ftp.disit.org). Tali dati vengono letti da un processo ETL che li inserisce all'interno del sistema.

4.2.3 Nuovo SW

È previsto un nuovo software su GUIDO che permetta lo scarico dei dati provenienti dai soli dispositivi TURMS inclusi in Sii-Mobility. L'operazione avviene trasferendo tramite protocollo FTP le stringhe NMEA provenienti dai TURMS corredate da un identificatore del veicolo.

La configurazione iniziale di TURMS, immediatamente successiva all'installazione, avviene con i consueti meccanismi previsti da GUIDO.

4.2.4 Installazione di TURMS

Il sistema di localizzazione satellitare TURMS ha le caratteristiche tecniche sotto indicate.

Si fa riferimento anche al doc. [Ref 3].



CARATTERISTICHE

Alimentazione 7-24VDC

Assorbimento 7mA* @ 12VDC Modalità LOW POWER

100mA @ 12VDC Modalità Operativa

150mA @ 12VDC GSM in TX

* dipendenti dalle attività GSM e dai comandi inviati

Modalità LOW POWER: Parte seriali non operative

GPS non operativo

GSM operativo

Batteria di backup

Batteria al Pb 2,3 A/h 12VDC Per alimentazione 24V

Batteria al Pb 1.3 A/h 6VDC per alimentazione 12V

Batteria Ni-Mh 270 mA/h 7,2Vdc Per alimentazione 12V

MECCANICHE

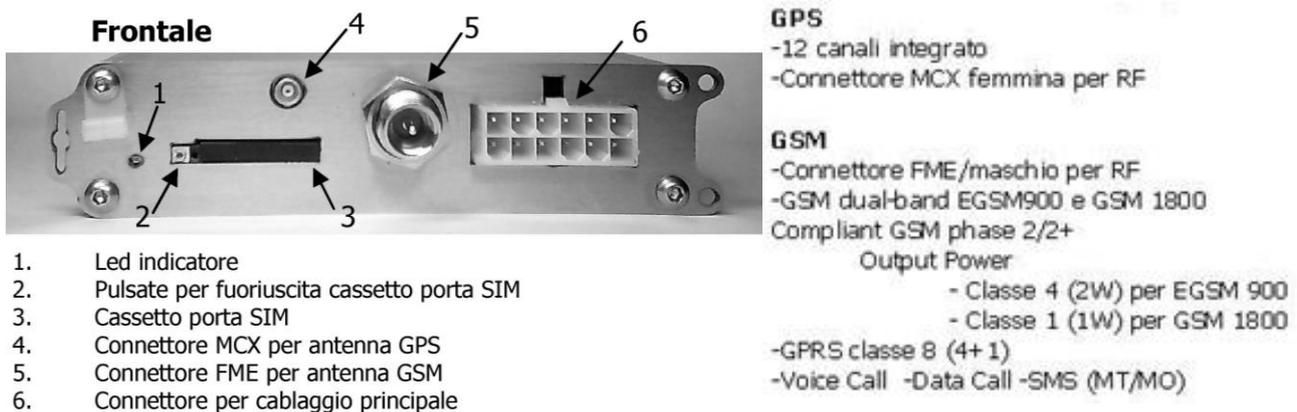
Involucro in alluminio anodizzato finiture in acciaio inox

Dimensioni 105mmx105mmx26mm

L'apparato TURMS non ha una interfaccia verso il conducente dell'autobus e può essere alloggiato nella parte anteriore del veicolo, ad esempio nella zona cruscotto.

Non è possibile, al momento, sviluppare l'installazione del TURMS in quanto occorre effettuare un esame tecnico degli autobus su cui sarà applicato. Questa verifica sarà fatta assieme ai gestori di TPL, con i quali si definiranno i montaggi, in accordo con le normative ed i regolamenti vigenti nell'ambito del trasporto pubblico.

I dettagli della installazione del kit a bordo saranno specificati nel relativo manuale [SiiM 10].



5 Kit carbus evoluto

Il *Kit carbus evoluto* è progettato per equipaggiare autobus di proprietà di gestori di TPL oppure auto di organizzazioni pubbliche o di privati cittadini.

Il Kit include un apparato HW/SW progettato e sviluppato appositamente per il progetto, che nel seguito viene riferito come *Piattaforma*. Si interfaccia fisicamente con *Sensori* installati sul veicolo.

Per lo sviluppo e il supporto del *Data Harvester* e della *Mobile App* si utilizza un *Tablet* di bordo, in modo da fornire un host espressamente pensato per applicazioni WEB based. Il Tablet è componente sostanziale del Kit.

La Piattaforma e il Tablet comunicano utilizzando gli apparati Bluetooth Low Energy di cui entrambi sono equipaggiati. Realizzano complessivamente un meccanismo di comunicazione che permette uno scambio di comandi / controlli / informazioni in modalità wireless.

5.1 Architettura generale

L'architettura generale, contestualizzata in Sii-Mobility, del Kit è mostrata in *Figura 23*.

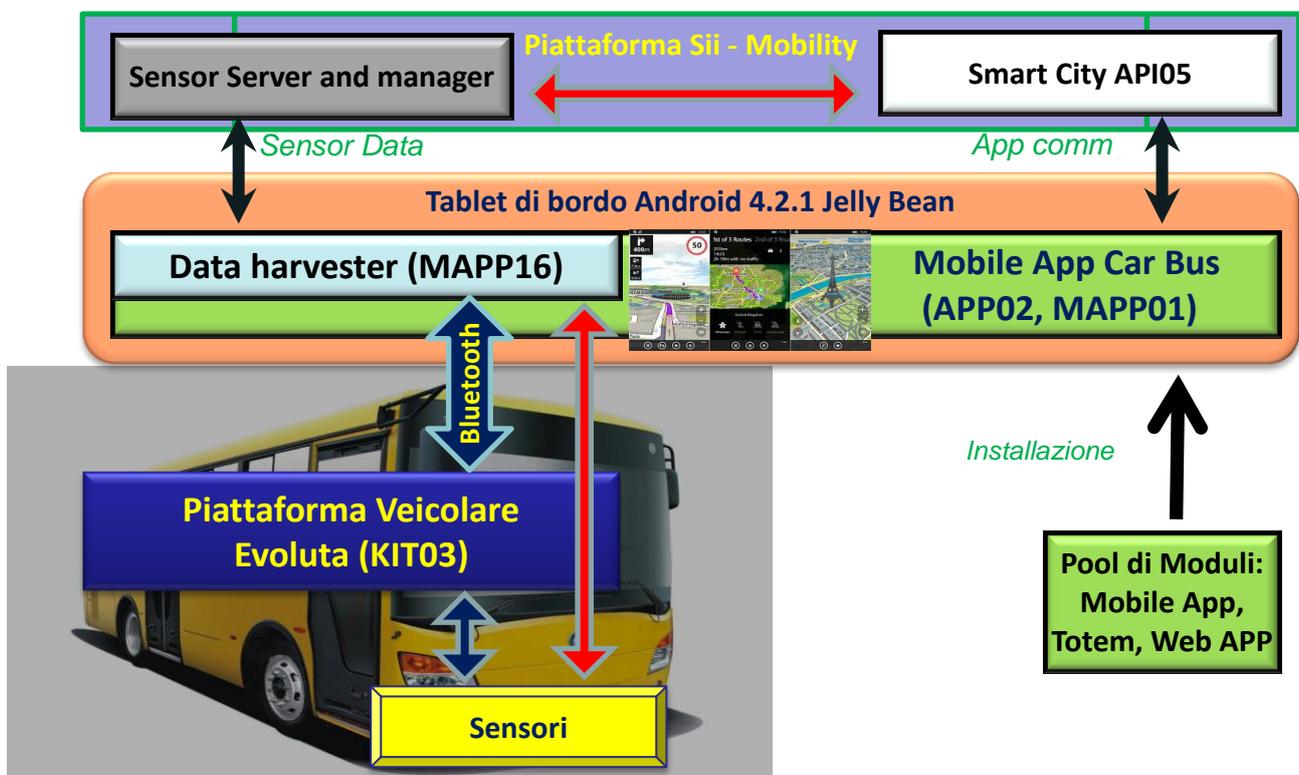


Figura 23 : Architettura generale del Kit carbus evoluto

I vantaggi offerti da questa organizzazione sono molteplici e rilevanti:

- chi sviluppa Data Harvester e Mobile App opera in un ambiente “familiare”, costituito da uno tablet commerciale, con caratteristiche ben standardizzate. Utilizza una installazione originale del sistema operativo Android, mentre un “porting” di Android sulla Piattaforma esporrebbe a rischi di comportamenti anomali,
- il canale di comunicazione Bluetooth può essere usato efficacemente sia per rendere disponibili al Data Harvester i dati dei sensori contenuti nella piattaforma stessa, sia per virtualizzare l’interazione con i sensori esterni alla piattaforma,

- la Piattaforma risulta essere sostanzialmente una estensione del Data Harvester, limitando di fatto il numero di partners coinvolti nella definizione e implementazione delle interfacce tra Data Harvester e Piattaforma e tra Data Harvester e sensori,
- la Piattaforma si basa su una architettura hardware “open” che costituisce uno *standard de facto* per applicazioni di questo tipo ([Ref 1]).

5.2 Specifica di dettaglio

5.2.1 Diagramma di contesto generale

La *Figura 24* mostra un primo livello di specifica e ripropone lo schema generale di architettura nella notazione utilizzata per la descrizione di dettaglio. A questo livello occorre definire con maggior precisione le interfacce mostrate.

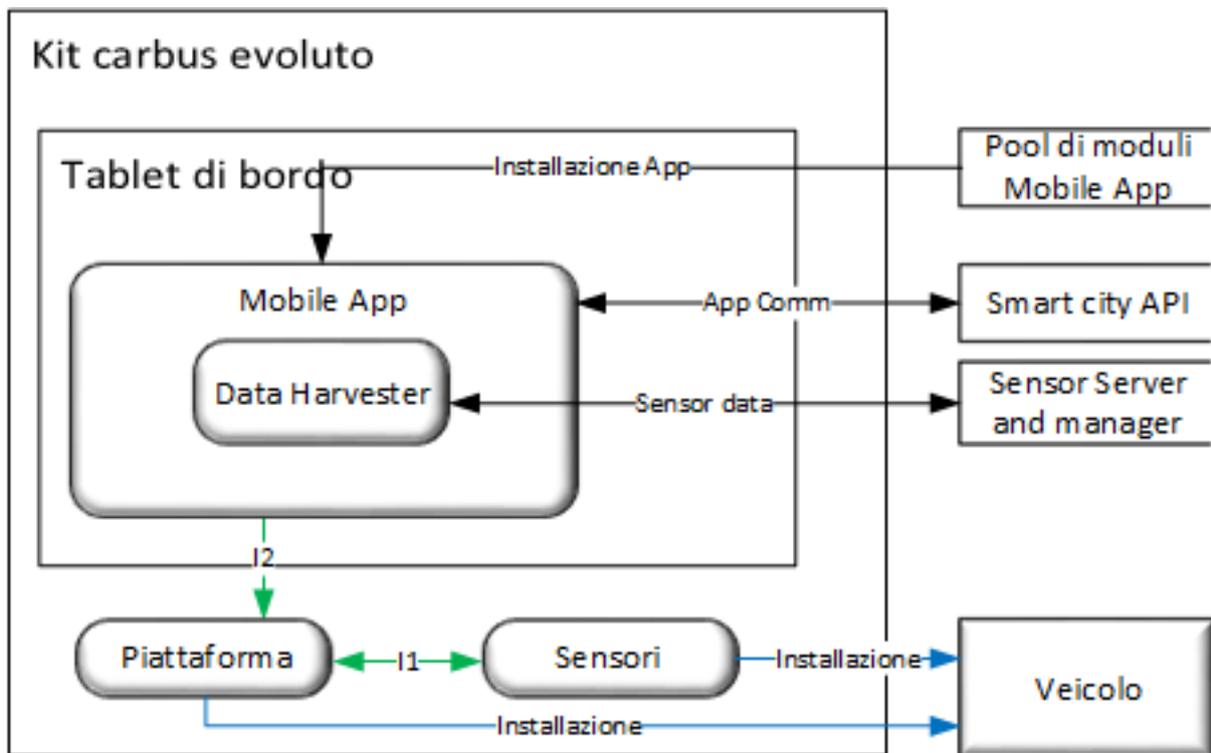


Figura 24 : Kit carbus evoluto - diagramma di contesto

In particolare:

- Interfacce esterne:
 - *Installazione App*, *App Comm* e *Sensor data* sono interfacce applicative realizzate con scambi di dati su WEB. Non vengono ulteriormente dettagliate in questo documento;
 - *Installazione* della Piattaforma e dei Sensori è un’interfaccia fisica che coinvolge la collocazione dei vari elementi. Richiede un momento di coordinamento tra i partners che realizzano i contenitori ed i supporti dei vari elementi del Kit e quelli che si occupano della installazione sui veicoli, propedeutica alla sperimentazione. L’argomento è sviluppato alla sezione 5.2.2.2 per la Piattaforma e nelle sezioni dedicate nel seguito ai vari Sensori;
- Interfacce interne, sviluppate nelle sezioni che seguono:

- I1 è l'interfaccia fisica e logica tra Piattaforma e sensori;
- I2 è l'interfaccia fisica e logica tra Data Harvester e Piattaforma;

5.2.2 Piattaforma, KIT03

5.2.2.1 Diagramma di contesto

La **Figura 25** mostra il diagramma di contesto della Piattaforma. Si fa distinzione tra sensori:

- *passivi*, per i quali l'interfaccia fisica è costituita da collegamenti elettrici e l'interfaccia logica dai livelli di tensione presenti sui collegamenti;
- *attivi*, che includono una CPU, per i quali l'interfaccia fisica è costituita da linee seriali e l'interfaccia logica dai messaggi scambiati secondo un certo protocollo.

I sensori attivi potranno anche avere collegamenti di comando / controllo analoghi a quelli dei sensori passivi in aggiunta allo scambio di messaggi.

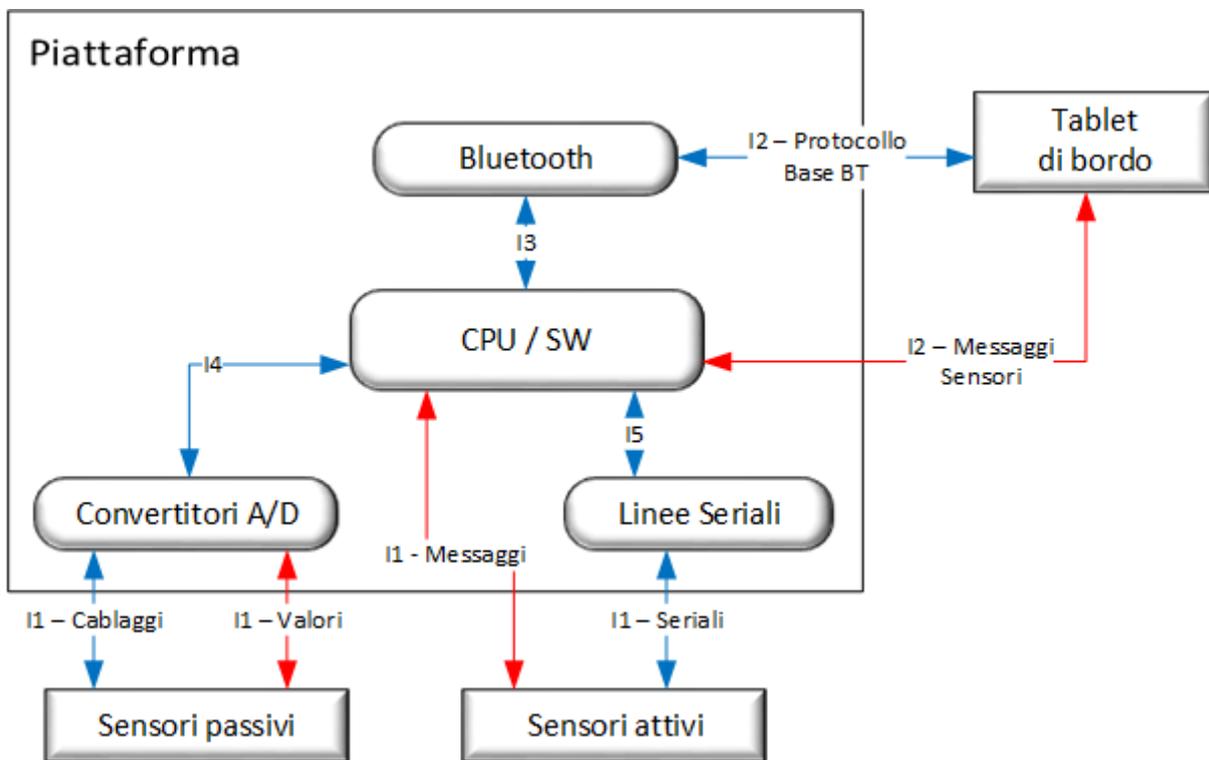


Figura 25 : Piattaforma – diagramma di contesto

La piattaforma include gli elementi seguenti, tutti descritti in dettaglio alla sezione 5.2.2.3:

- convertitori Analogico/Digitale e Digitale/Analogico per la gestione dei collegamenti elettrici con i sensori passivi e attivi. Hanno riscontro nelle porte di I/O della motherboard;
- linee seriali per lo scambio di messaggi con i sensori attivi. Hanno riscontro nelle porte di I/O della motherboard;
 - una scheda CPU (motherboard) che include un dispositivo Bluetooth Low Energy per le comunicazioni con lo smartphone;

I sensori e apparati interfacciati sono:

- uno “Smart Node” che raccoglie i messaggi dalle telecamere presenti a bordo:

- telecamera per conteggio passeggeri (5.2.4.2),
 - telecamera per monitoraggio traffico e ambientale (5.2.4.4),
- e li fornisce alla Piattaforma su linea seriale;
- un CanBus Sniffer che acquisisce i messaggi in transito sul bus del veicolo, descritto in dettaglio alla sezione 5.2.4.1
 - sensori ambientali, descritti in dettaglio alla sezione 5.2.4.5

Principali attività e interfacce coinvolte:

- ❖ **gestione sensore passivo** : *I1 – Cablaggi* rappresenta i collegamenti cablati con il sensore mentre *I1 – Valori* rappresenta il significato della grandezza (tipicamente una tensione) che si acquisisce da ciascun collegamento, p.es. proporzionale alle parti per milione di un agente inquinante, oppure proporzionale alla distanza del veicolo che segue. La piattaforma acquisisce i valori, li codifica in messaggi secondo le convenzioni definite per *I2 – protocollo Base BT* e li invia allo smartphone (Data Harvester). È importante notare che viene trasmesso il dato grezzo, il suo significato (interfaccia logica) non è rilevante per la piattaforma, mentre lo è per il Data Harvester. L'interfaccia *I4* rappresenta le convenzioni, locali alla piattaforma (ECM-ELFI), sulla locazione dei convertitori e sulla loro gestione;
- ❖ **gestione sensore attivo** : i messaggi dai sensori (*I1 - Messaggi*) sono ritrasmessi senza modifiche allo smartphone e viceversa secondo le convenzioni definite per *I2 – protocollo Base BT*. L'interfaccia *I1 – Seriali* rappresenta solo il tipo di linea seriale, mentre *I5* rappresenta le convenzioni, locali alla piattaforma (ECM-ELFI), sulla locazione dei buffer e sulla gestione degli eventi TX/RX. Se un sensore attivo ha anche collegamenti cablati, questi sono gestiti come nel caso precedente;
- ❖ **le interfacce fisiche** con i sensori sono condivise tra chi sviluppa il sensore (che le definisce) e chi sviluppa l'HW della piattaforma (ELFI) che le recepisce;
- ❖ **le interfacce logiche** con i sensori sono condivise tra chi sviluppa il sensore (che le definisce) e chi sviluppa il Data Harvester (DISIT) che le recepisce;
- ❖ **gestione del collegamento Bluetooth** : il SW della piattaforma e quello del data harvester definiscono e condividono le informazioni necessarie per realizzare un canale di comunicazione Bluetooth (interfaccia *I2 – protocollo Base BT*) con un suo protocollo. Utilizzano, rispettivamente, i servizi della piattaforma, interfaccia *I3*, e di Android (vedere [Figura 29](#)). Il protocollo comprende il formato dei messaggi (header, footer, etc.), messaggi di controllo e messaggi di dati. Nei messaggi dati sono contenuti (senza modifiche) i messaggi da e per sensori attivi e inoltre i messaggi generati dalla piattaforma per i sensori passivi.

Dunque l'interfaccia logica *I2 – Messaggi sensori* è costituita dall'OR di:

- *I1 – Valori*
- *I1 - Messaggi*

5.2.2.2 Meccanica, contenitori, carpenterie, installazione a bordo

Al momento non è possibile sviluppare questo aspetto, che richiede analisi congiunta con i gestori di TPL per individuare soluzioni funzionali al progetto, in linea con leggi e regolamenti vigenti e compatibili con forma e dimensioni del vano in cui potrà essere installato il contenitore.

In prima istanza, si può assumere che il contenitore del kit non sarà visibile in quanto contenuto nel cruscotto dell'autobus e non avrà sensori o attuatori al suo interno ma soltanto la piattaforma. Avrà quindi esclusivamente funzione di contenimento e protezione.

Per il Tablet di bordo occorrerà invece concordare coi gestori TPL posizionamento e fissaggio più opportuni, trovando contestualmente un valore ragionevole per le sue dimensioni.

I dettagli della installazione del kit a bordo sono specificati nel relativo manuale [SiiM 10].

5.2.2.3 Hardware

5.2.2.3.1 Motherboard con Bluetooth Low Energy

Per il kit carbus evoluto è stata selezionata la stessa motherboard (Arduino 101) del kit bike. La potenza di calcolo e la disponibilità di risorse, superiori rispetto alle precedenti schede Arduino, permettono di gestire anche questo caso applicativo.

Per la descrizione della motherboard si vedano quindi le sezioni 3.2.2.3.1 e 3.2.2.3.2.

5.2.2.3.2 Interfacciamento dei sensori

La motherboard interagisce con i sensori descritti nelle sezioni seguenti: ovvero

- telecamere intelligenti per il conteggio dei passeggeri e per il monitoraggio del traffico (5.2.4.3, 5.2.4.4) tramite l'apparato Smart Node (5.2.4.2),
- sensori ambientali (5.2.4.5),
- un modulo CanBus sniffer (5.2.4.1).

Per quanto riguarda le telecamere intelligenti, come prospettato in [SiiM 4] sezione 5, si prevede un unico elemento (Smart Node) che raccoglie ed elabora i dati di tutte le telecamere presenti a bordo e fornisce alla piattaforma del kit i risultati delle elaborazioni per la trasmissione verso il Data Harvester. Per lo Smart Node si riserva la linea seriale più performante, I²C.

Il CanBus sniffer richiede a sua volta una linea seriale, si prevede di utilizzare le porte RX e TX.

L'interfacciamento dei sensori ambientali è simile quello del kit bike, tranne il fatto che a bordo autobus i sensori sono più numerosi e quindi vengono utilizzate più porte di ingresso analogiche. Anche in questo caso richiedono dell'hardware per l'adattamento dei segnali, contenuto in un Modulo di condizionamento. Ci si propone di realizzare una proto-shield aggiuntiva dove posizionare detto HW.

Le alimentazioni dei sensori saranno fornite tutte dall'impianto elettrico dell'autobus e non dalle porte della piattaforma: questa è una soluzione che garantisce maggiore robustezza e diminuisce il rischio di interferenze nel funzionamento di tutto il kit. La Figura 26 e la Tabella 8 mostrano uno schema generale del kit e il dettaglio del collegamento delle interfacce fisiche alle porte di Arduino 101.

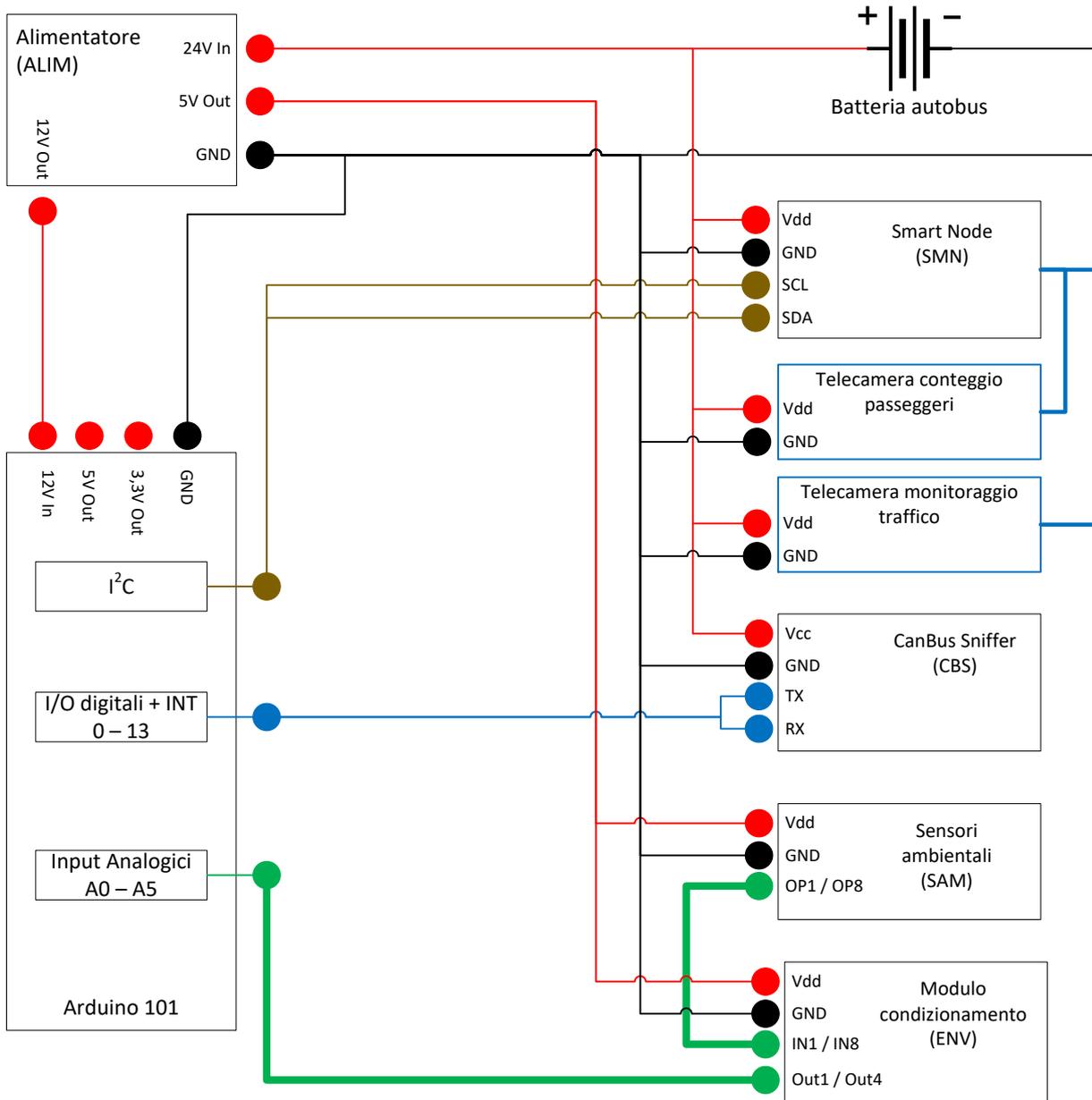


Figura 26 : Schema generale del kit carbus evoluto

Kit Carbus Evoluto - Connessioni fisiche tra piattaforma e sensori							
Analog IN	A0	A1	A2	A3	A4	A5	
	ENV Out1	ENV Out2	ENV Out3	ENV Out4			
I/O/INT Ports	0/RX	1/TX	2°	3*	4	5*°	6*°
	CBS RX	CBS TX					
I/O/INT Ports	7	8°	9*	10°	11°	12°	13°

Power	ATN/SS	IOREF	RESET	3,3	5	GND	Vin
						Ground	12V
Other	SCL	SDA	AREF	ICSP/ MISO	ICSP/ MOSI	ICSP/ SCK	ICSP
	SMN SCL	SMN SDA					

Tabella 8: Kit carbus evoluto - connessioni fisiche tra motherboard e sensori

*: porte che supportano PWM

°: porte che supportano interrupt sui cambiamenti di stato

5.2.2.4 Software

In questo paragrafo è descritto il design del firmware della Piattaforma, denominato Carbus Kit Manager (CKM). La Piattaforma è dotata di firmware che gestisce i Sensori (I1) installati sul veicolo oltre che la comunicazione Bluetooth (I2) con il Tablet di bordo.

5.2.2.4.1 Casi d'uso

In Figura 27 è mostrato il digramma UML dei casi d'uso (CU) in cui sono descritti gli scenari di utilizzo della Piattaforma da parte degli attori.

NOTE: Per la natura del mezzo, si assume che il caso d'uso Vehicular association (UC1) sia attivato una tantum nella fase di installazione del Kit Carbus Evoluto sul mezzo, ovvero nella fase di primo utilizzo dell'apparato. Il caso d'uso Vehicular deassociation (UC5) è attivato nel caso in cui il sistema Kit Carbus Evoluto viene disinstallato dal mezzo, esempio: manutenzione.

Indipendentemente dal mezzo in cui il Kit Carbus Evoluto è installato si hanno i seguenti casi d'uso:

- **Vehicular association:** coinvolge l'interfaccia I2. Il presente UC è necessario per attivare gli altri UC del CKM. Il caso d'uso è attivato una tantum nella fase di installazione del Kit Carbus Evoluto:
 - **ID:** UC1;
 - **Attori:** Tablet, Piattaforma
 - **Pre-condizione:**
 1. Il Tablet è attivo con installate la Mobile App ed il Data Harvester;
 2. La Piattaforma è funzionante;
 3. Il modulo Bluetooth della Piattaforma e del Tablet è attivo;
 - **Sequenza degli eventi:** secondo le specifiche del Protocollo base BT (Sezione 5.2.4):
 1. Il caso d'uso inizia quando la Piattaforma riceve un messaggio di associazione da parte del Tablet utente;
 2. La Piattaforma invia al Tablet un messaggio di risposta alla richiesta di associazione;
 - **Post-condizione:** il Tablet e la Piattaforma sono associati e gli altri casi d'uso sono attivi.
- **Data gathering:** coinvolge l'interfaccia I1. Acquisizione periodica dei dati di misura dei sensori installati:
 - **ID:** UC2;
 - **Attori:** Time, Piattaforma, Sensor platform;
 - **Pre-condizione:** UC1 completato;
 - **Sequenza degli eventi:**
 1. Il CU inizia quando scade il Timer periodico di gathering;
 2. La Piattaforma legge il valore del sensore;
 3. La Piattaforma esegue l'operazione di Timestamping (e georeferenziazione(?)) della misura;
 4. La Piattaforma imposta il Timer periodico di gathering;
 - **Post-condizione:** la Piattaforma ha acquisito il dato del sensore ed il Timer periodico è attivo.
- **Data sending:** coinvolge l'interfaccia I2. Invio dei dati al Tablet:
 - **ID:** UC3;

Figura 27 : Carbus Kit Manager Bike – casi d'uso

5.2.2.4.2 Macchina a stati

In Figura 28 è riportata la macchina a stati del CKM del Kit Carbus Evoluto. L'evento `switchON` rappresenta l'azione di accensione della Piattaforma a cui segue la fase di `setup()` dell'hardware. Il CKM si trova nello stato di `Idle` in cui il CKM è attivo e non associato, dunque è in attesa di una richiesta di associazione da parte del Tablet di bordo. In questo stato il CKM risponde ai seguenti eventi:

- `reset` a cui segue l'azione di `setup()` dell'hardware;
- `swichOFF` che attiva lo stato finale `OFF`;
- `associationRequest` seguito dall'azione `tryAssociation()` che esegue le operazioni di associazione. Se la procedura di associazione con il Tablet va a buon fine, il CKM entra nel superstato `Associated`, viceversa torna nello stato di `Idle`.

Descrizione superstato Associated

Quando la Piattaforma è associata a un Tablet lo stato `Associated.Synchronization` è attivo e l'azione di sincronizzazione dei clock `sendSyncRequet()` è eseguita. Quando termina la fase di sincronizzazione, evento `receivedSyncTime`, viene eseguita l'azione `setTimers()` che impostano tutti i Timer periodici e lo stato `Associated.Executing` è attivato. In questo stato il CKM risponde ai seguenti eventi:

- `upload`: invio dei messaggi con i dati dei sensori. Azione eseguita `sendData()`;
- `gathering`: lettura dei valori di misura dei sensori. Azione eseguita `readSensors()`;
- `notSync`: è necessaria la sincronizzazione dei clock. Lo stato `Associated.Synchronization` è attivo;
- `deassociationRequest`: arriva una richiesta di deassociazione da parte del Tablet. Il CKM torna nello stato `Idle`;
- `switchOFF`: spegnimento del dispositivo. Stato finale `OFF`.

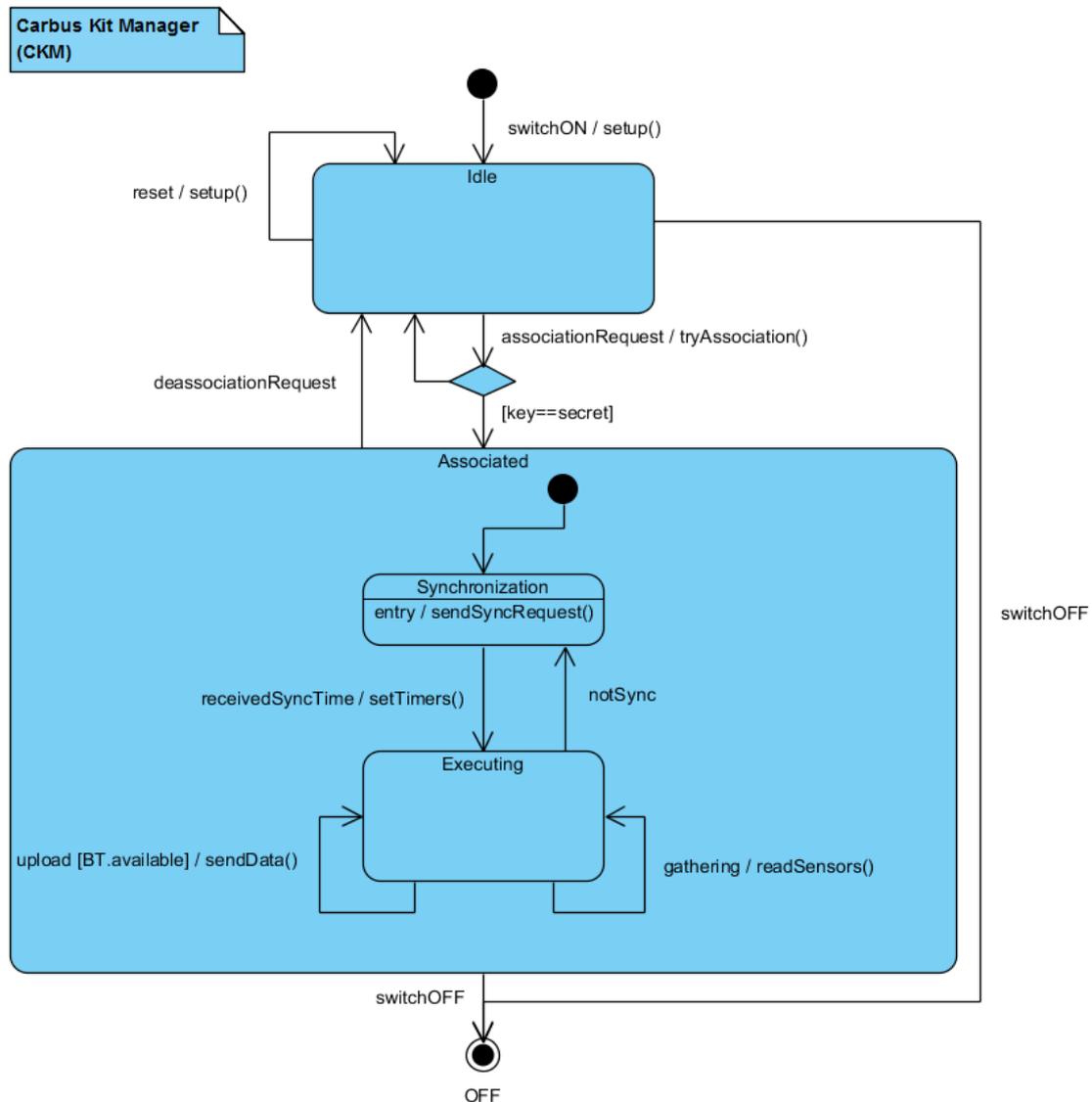


Figura 28: Carbus Kit Manager Bike – FSM

5.2.2.4.3 Moduli software

Complessivamente, il software del CKM si articola nei moduli:

- **Startup** – esegue l’inizializzazione della Piattaforma e di tutti i sensori collegati;
- **Controllore** – implementa la macchina a stati descritta in 5.2.2.4.2 per realizzare i casi d’uso descritti in 5.2.2.4.1. Attiva i moduli software descritti di seguito con una politica periodica oppure event-driven a seconda dei casi;
- **Gestione Bluetooth** – realizza il comando e controllo dell’hardware di comunicazione Bluetooth;
- **Protocollo base Bluetooth** – implementa, lato Piattaforma, il protocollo di comunicazione con il Data Harvester, (si veda la Sezione 5.2.5 per dettagli);
- **Gestione Telecamere** - realizza il comando e controllo dei sensori attivi su linea seriale – fornisce al controllore i servizi per l’acquisizione dei dati;
- **Gestione sensori ambientali** - realizza il comando e controllo dei sensori passivi – fornisce al controllore i servizi per l’acquisizione dei dati;

- **Gestione CANbus sniffer** - realizza il comando e controllo dei sensori attivi – fornisce al controllore i servizi per l’acquisizione dei dati;
- **Gestione linee seriali** – modulo di servizio per tutti gli altri moduli;
- **Gestione segnali analogici** – modulo di servizio per tutti gli altri moduli.

5.2.3 Tablet di bordo

5.2.3.1 Diagramma di contesto

La *Figura 29* mostra il diagramma di contesto dello Smartphone Utente

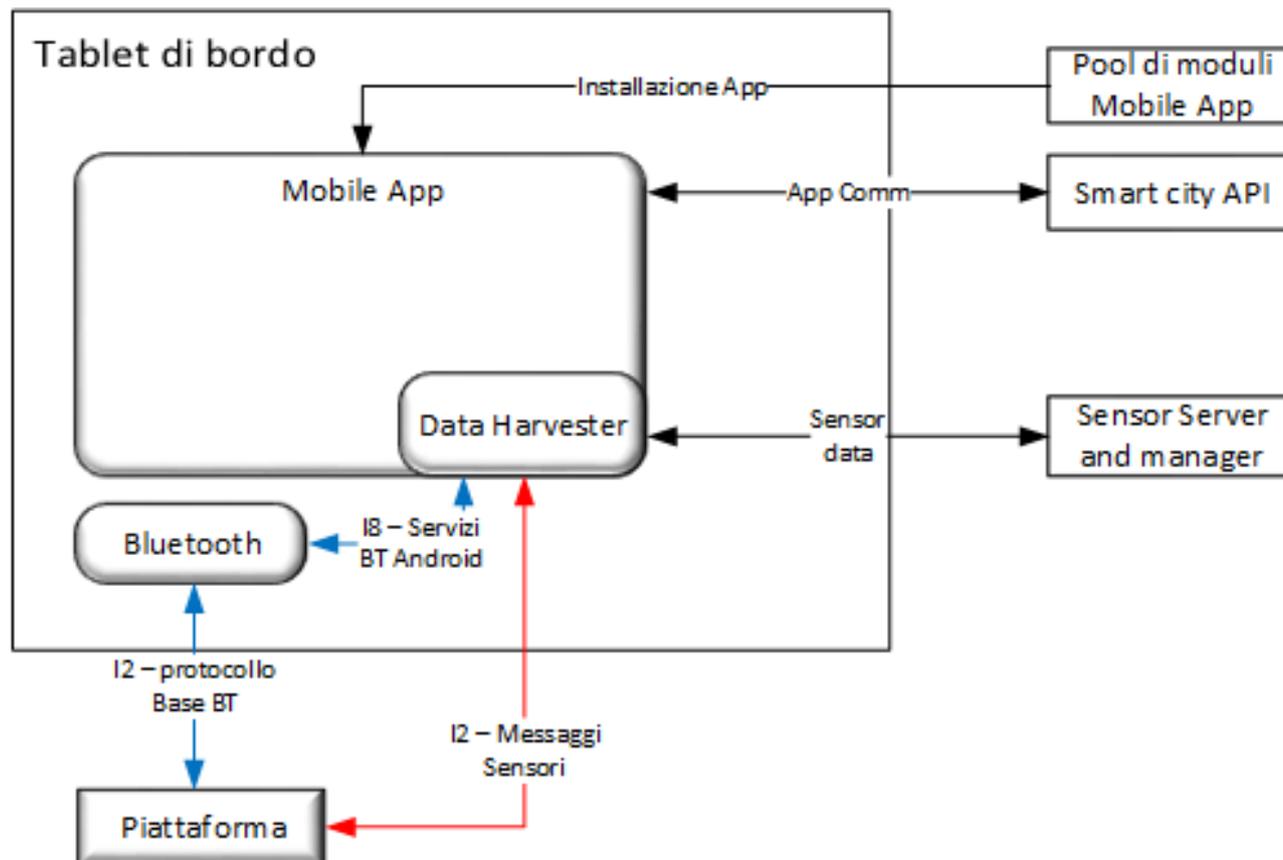


Figura 29 : Smartphone – diagramma di contesto

Di seguito sono indicate le caratteristiche minime che lo Smartphone dell’Utente deve possedere per supportare efficacemente gli elementi del Kit che vi sono installati:

- ✚ Sistema Operativo **Android 4.2.1 Jelly Bean** o revisioni successive retro-compatibili;
- ✚ Bluetooth Low Energy
- ✚ GPS
- ✚ Accelerometri

5.2.3.2 Data Harvester, MAPP16

Il Data Harvester (MAPP16) nelle sue varie istanze e versioni è descritto in altri documenti che al momento non sono determinati.

5.2.3.3 Mobile App Car Bus, APP02, MAPP01

Le Mobile Apps APP02 e MAPP01 sono descritte in altro documento di progetto, che al momento non è determinato.

APP02 incorpora MAPP01 che esegue il monitoraggio di situazioni critiche analizzando i dati accelerometrici del Tablet di bordo. MAPP01 è un modulo progettato da MIZAR e descritto nella sezione seguente.

5.2.3.4 Sensori per situazioni critiche (SN02)

In conformità con quanto stabilito nella specifica di progetto (DE1.2) ed in conformità con i requisiti di progetto, il monitoraggio di situazioni critiche verrà realizzato attraverso l'elaborazione del dato di accelerazione rilevato a bordo, seguito dalla generazione dei relativi allarmi.

Per quanto condiviso in sede di riunione plenaria di progetto (si veda [SiiM 6], slide 70), il relativo modulo Sii-Mobility SN02 risulta abolito in quanto il sensore utilizzato per l'identificazione di situazioni critiche sarà l'accelerometro integrato nel dispositivo mobile presente a bordo (Tablet di bordo Android 4.2.1). Il modulo MAPP01 associato conterrà la logica per l'elaborazione di tale segnali e relativa generazione allarmi, e sarà parte appartenente all'APP02, come da architettura descritta all'inizio del Capitolo 5 del presente documento.

5.2.4 Sensori

La maggior parte dei sensori collegati alla Piattaforma sono prodotti nel contesto dell'Attività 3.2 e di seguito sono forniti i loro riferimenti nel deliverable [SiiM 4] dove è sviluppata la loro specifica di dettaglio.

Fa eccezione il CANbus Sniffer, che è un apparato già esistente (un Tool nella terminologia utilizzata in [SiiM 2]) fornito da ELFI, che viene descritto direttamente in questo documento.

Per tutti i Sensori è presente anche una sezione destinata a contenere spiegazioni e suggerimenti per l'installazione a bordo veicolo.

5.2.4.1 CanBus Sniffer

CAN (Controller Area Network) e OBD-II (On Board Diagnostics) sono due protocolli standard di comunicazione.

Per ricevere dati relativi allo stato del veicolo si collega un'interfaccia OBD-II (On Board Diagnostics) al CAN-bus (la linea di trasporto dei dati che segue protocollo di invio e ricezione CAN, Controller Area Network). Per come è studiato il CAN, quando una centralina invia un messaggio sul bus tale messaggio viene ricevuto da tutte le altre centraline collegate.

L'OBD-II è uno standard che è diventato obbligatorio sulle automobili a partire dal 1996 negli Stati Uniti e dal 2001 in Europa (EOBD), definito per rendere omogeneo il modo di codificare i dati di diagnostica a bordo veicolo.

Oltre a definire lo standard di comunicazione e una serie di comandi standard per la richiesta delle informazioni, questo protocollo definisce anche il tipo di connettore, ovvero la porta di collegamento fisico al bus CAN; il connettore prevede anche una linea che fornisce energia al dispositivo collegato (anche a motore spento), prendendola dalla batteria del veicolo. Lo standard definisce anche una lista di parametri del veicolo da monitorare e come codificare ciascun dato relativo, compresi alcuni codici di errore, divisi secondo le aree di appartenenza.

La connessione fisica al bus CAN, permette di ascoltare tutti i messaggi che le centraline inviano sul bus, ma anche dialogare attivamente con le centraline, richiedendo dei dati per esempio, oppure

inviando dei comandi. Quest'ultimo punto rende delicata da un punto di vista normativo e della sicurezza dei veicoli in movimento, il collegamento con questa linea. Per questa ragione tutto ciò che segue deve essere valutato nella sua applicazione effettiva in relazione sia al quadro legislativo vigente che a una più ampia e prudentiale valutazione in termini di sicurezza dei dispositivi installati.

5.2.4.1.1 Interfaccia fisica

Il collegamento fisico al veicolo avviene usando un opportuno connettore OBD-II alla porta OBD presente a bordo del veicolo.

Il connettore OBD-II, prevede due interfacce standard hardware tipo A (per veicoli con batterie a 12V) e tipo B (per veicoli con batterie a 24V), entrambi sono a 16 piedi, ma con forma leggermente diversa (Figura 30).



Figura 30 : Connettori OBD-II – tipo “a” (sinistra) e tipo “b” (destra)

Al connettore a questo punto arrivano tutti i dati presenti sul bus CAN.

Si tratta a questo punto di rendere disponibili questi dati al Data Harvester di Sii-Mobility e a tale fine si può procedere in diversi modi: nella scelta della modalità più opportuna sono da valutare questioni tecniche ma soprattutto, come detto in precedenza, questioni relative alla normativa vigente in merito di sicurezza su veicoli in movimento.

Una prima opzione è quella di connettere la porta OBD-II alla Piattaforma Arduino tramite opportuno adattatore o shield dedicata. Questa opzione ha il vantaggio di rendere direttamente disponibili alla piattaforma Arduino tutti i dati del CAN-bus, che può quindi essere interrogato secondo necessità, inviando poi i dati al Data Harvester via Bluetooth insieme ai dati di tutti gli altri sensori.

Esistono adattatori con porte UART o I²C, mentre le shield più diffuse comunicano via SPI. Per non occupare spazio con una ulteriore shield si prevede di impiegare un adattatore con uscite TTL, per esempio l'adattatore Freematics OBD-II UART Adapter V1 (for Arduino), con porta di tipo A, e che supporta vari protocolli CAN (CAN 500Kbps/11bit, CAN 500Kbps/29bit, KWP2000 Fast, KWP2000 5Kbps), previa verifica del tipo di porta OBD-II effettivamente presente sugli autobus (tipo A e tipo B), e il protocollo CAN impiegato (Figura 31).

L'adattatore è provvisto di libreria dedicata (vedi Interfaccia Logica), supporta i comandi ELM327 AT, con una frequenza di acquisizione dei dati fino a 100Hz.



Figura 31 : UART Adapter per CanBus Sniffer

È ovvio che il software di Arduino si asterrà dall'eseguire qualsiasi scrittura verso il CanBus, salvo quelle eventualmente necessarie per la configurazione/attivazione. Dal momento che sarà un open-source questa proprietà sarà verificabile da parte di chiunque.

Tuttavia non si ritiene che questa sia garanzia sufficiente per la sicurezza dei veicoli in condizioni di normale esercizio e pertanto si potrà attivare la funzione solo in un'area controllata, istruendo opportunamente il conducente e ovviamente senza passeggeri a bordo. Per tutte le altre prove di sperimentazione sarà sufficiente disconnettere lo Sniffer dal CanBus.

Volendo andare oltre, nel contesto di Sii-Mobility, occorre individuare un dispositivo fornito di certificazione/omologazione che garantisce la non intrusività sul CanBus. Una ricerca in questo senso è già in corso e proseguirà attivamente. Si potranno trovare:

- dispositivi che garantiscono intrinsecamente la non intrusività in quanto fisicamente privi della possibilità di scrittura sul CanBus. In questo caso potranno essere connessi alla Piattaforma come prospettato;
- dispositivi che forniscono anche un software di gestione, a sua volta certificato, che presenta i dati di telemetria ma non consente operazioni potenzialmente pericolose. Con tutta probabilità utilizzeranno comunicazioni wireless e forniranno una APP per dispositivo mobile come interfaccia. In questo caso la APP dovrebbe essere caricata sul Tablet di servizio e opportunamente interfacciata con il Data Harvester per il trasferimento dei dati sulla piattaforma Sii-Mobility.

Data la grande varietà di dispositivi presenti in commercio, ciascuno con le sue specifiche, non è possibile in questa fase dettagliare ulteriormente questo possibile scenario. Pertanto nel seguito ci riferiamo al primo caso, quello in cui si collega direttamente il CanBus Sniffer alla Piattaforma Arduino.

5.2.4.1.2 Interfaccia logica

La Piattaforma riceve via protocollo seriale i dati del CAN-bus.

L'adattatore proposto, il Freematics OBD-II ha disponibile una libreria Arduino che rende possibile configurare il dispositivo e prevede alcune API per interrogare facilmente i dati attraverso comandi definiti secondo protocollo ELM327.

Esiste uno standard per definire i metodi di richiesta dei diversi dati diagnostici e una lista di parametri standard disponibili normalmente dalla centralina di controllo. I parametri sono identificati da uno specifico parametro (PID).

P00xx – Fuel and air metering and auxiliary emission controls.

P01xx – Fuel and air metering.

P02xx – Fuel and air metering (injector circuit).

P03xx – Ignition system or misfire.

P04xx – Auxiliary emissions controls.

P05xx – Vehicle speed controls and idle control system.

P06xx – Computer output circuit.

P07xx – Transmission.

P08xx – Transmission.

I dati di ciascuno PID sono mandati uno dopo l'altro, e il tempo con cui sono inviati dipende dalle caratteristiche della centralina e dal suo stato (se più o meno occupata): con le centraline odierne tipicamente i dati sono mandati fino a 50 volte al secondo.

5.2.4.1.3 Installazione a bordo

L'installazione a bordo dipende dal tipo di soluzione scelta, dopo la valutazione delle garanzie rispetto alla normativa vigente e alla sicurezza dei veicoli in movimento. In ogni caso si tratta di collegarsi al CAN-bus attraverso la porta OBD-II presente sull'autobus, attraverso opportuno connettore. Dall'altro lato di questo connettore può poi trovarsi la piattaforma Arduino (connessa attraverso le porte seriali), oppure il Tablet di servizio che comunica via cavo o wireless con il dispositivo di connessione alla porta OBD-II.

La soluzione ottimale sarà individuata a seguito di una analisi approfondita, coinvolgendo in particolare i Partners gestori di TPL, e specificata nel relativo manuale [SiiM 10] a cura di CReAI.

5.2.4.2 Smart Node, SN04

Lo **Smart Node** è in grado di raccogliere dati da un certo numero di Sensori Ottici. L'elaborazione dell'immagine può essere distribuita tra Sensori Ottici e Smart Node oppure no, ma in ogni caso è confinata a questo livello: dallo Smart Node escono solo informazioni sintetiche sotto forma di brevi messaggi che vengono trasmessi a una Piattaforma basata su Arduino (esempio: numero di passeggeri, segnalazione di coda, segnalazione di evento del traffico, targa di un veicolo)

Questo apparato è descritto nel deliverable dedicato ai sensori innovativi ([SiiM 4] e/o [SiiM 13]) alla sezione 5.

5.2.4.2.1 Installazione a bordo

Al momento non è possibile sviluppare questo aspetto, che richiede analisi congiunta con i gestori di TPL per individuare soluzioni funzionali al progetto e in linea con leggi e regolamenti vigenti. Le modalità di installazione a bordo saranno specificate nel relativo manuale [SiiM 10] a cura di PROJECT.

5.2.4.3 Telecamera Intelligente per conteggio passeggeri, SN05

I sensori video sono delle speciali telecamere da installare a bordo di mezzi pubblici (kit veicolari evoluti) od in tratti stradali da controllare, dedicati a molteplici impieghi legati alla gestione ed alla sicurezza a bordo e del traffico circostante. Gli algoritmi previsti consentono la classificazione degli eventi in modo da consentire una rapida reazione sia del mezzo che degli enti preposti alla gestione del traffico cittadino.

In questo caso, l'apparato è specializzato per il conteggio dei passeggeri a bordo autobus.

Questo apparato è descritto nel deliverable dedicato ai sensori innovativi ([SiiM 4] e/o [SiiM 13]) alla sezione 2.2.1.

5.2.4.3.1 Installazione a bordo

Al momento non è possibile sviluppare questo aspetto, che richiede analisi congiunta con i gestori di TPL per individuare soluzioni funzionali al progetto e in linea con leggi e regolamenti vigenti. Le modalità di installazione a bordo saranno specificate nel relativo manuale [SiiM 10] a cura di PROJECT.

5.2.4.4 Telecamera Intelligente per monitoraggio traffico e ambientale, SN05

I sensori video sono delle speciali telecamere da installare a bordo di mezzi pubblici (kit veicolari evoluti) od in tratti stradali da controllare, dedicati a molteplici impieghi legati alla gestione ed alla sicurezza a bordo e del traffico circostante. Gli algoritmi previsti consentono la classificazione degli eventi in modo da consentire una rapida reazione sia del mezzo che degli enti preposti alla gestione del traffico cittadino.

In questo caso, l'apparato è specializzato per il monitoraggio del traffico e ambientale.

Questo apparato è descritto nel deliverable dedicato ai sensori innovativi ([SiiM 4] e/o [SiiM 13]) alla sezione 2.2.4.

5.2.4.4.1 Installazione a bordo

Al momento non è possibile sviluppare questo aspetto, che richiede analisi congiunta con i gestori di TPL per individuare soluzioni funzionali al progetto e in linea con leggi e regolamenti vigenti. Le modalità di installazione a bordo saranno specificate nel relativo manuale [SiiM 10] a cura di PROJECT.

5.2.4.5 Sensori ambientali

Si tratta di sensori dedicata alla valutazione della qualità dell'aria attraverso la misurazione di alcuni parametri che, nello specifico, verranno poi decisi in sede di ulteriore avanzamento del progetto.

Il sistema sarà compatibile con la rilevazione delle seguenti grandezze:

- NO₂
- NO
- O₃
- H₂S
- SO₂
- CO
- VOC (Volatile Organic Compound)

Che rappresentano i principali inquinanti

Poiché i sensori per il loro corretto funzionamento richiedono di essere alimentati con tensioni stabili devono essere montati su delle board dedicate che prendono il nome di AFE (Analog Front End) che prevedono differenti possibili congiurazioni,

Riportiamo di seguito delle tabelle che indicano come i sensori potranno essere combinati nelle varie tipologie di AFE:

SN1	SN2
NO ₂	O ₃
NO ₂ or O ₃	NO
NO ₂ or O ₃	CO or SO ₂ or H ₂ S
NO	CO or SO ₂ or H ₂ S
CO	SO ₂ or H ₂ S

2 sensor AFE

SN1	SN2	SN3	PID
NO ₂	O ₃	CO or SO ₂ or H ₂ S	PID-AH
NO ₂	O ₃	NO	PID-AH
NO ₂	CO	SO ₂ or H ₂ S	PID-AH
NO ₂	CO or SO ₂ or H ₂ S	NO	PID-AH
CO	SO ₂	H ₂ S	PID-AH

3 Sensors + PID AFE

SN1	SN2	SN3
NO ₂	O ₃	CO or SO ₂ or H ₂ S
NO ₂	O ₃	NO
NO ₂	CO	SO ₂ or H ₂ S
NO ₂	CO or SO ₂ or H ₂ S	NO
CO	SO ₂	H ₂ S

3 sensor AFE

SN1	SN2	SN3	SN4
NO ₂	O ₃	CO or SO ₂ or H ₂ S	CO or SO ₂ or H ₂ S
NO ₂	O ₃	NO	CO or SO ₂ or H ₂ S
NO ₂	CO	SO ₂ or H ₂ S	SO ₂ or H ₂ S
NO ₂	CO or SO ₂ or H ₂ S	NO	CO or SO ₂ or H ₂ S
CO	SO ₂	NO	H ₂ S

4 Sensors AFE

5.2.4.5.1 Interfacce

Questo sensore è descritto nel documento [SiiM 4]. In particolare:

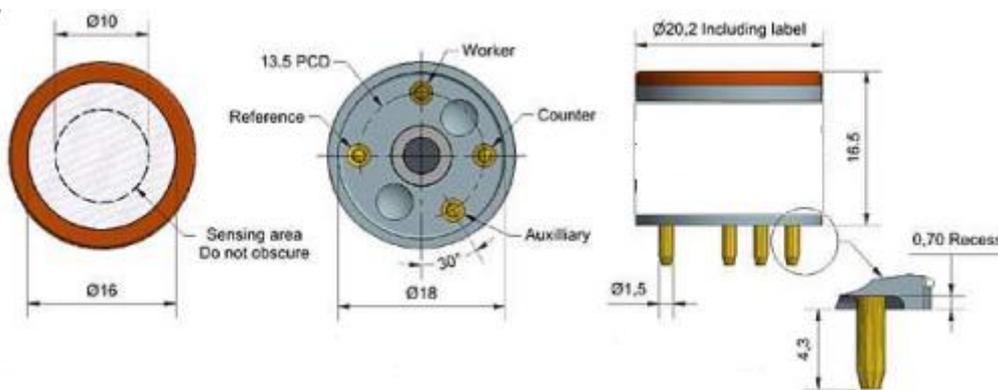
- l'interfaccia fisica è descritta alla sezione 3.1.1.2 e 3.2.3.2.
- l'interfaccia logica è descritta alla sezione 3.1.1.3 e 3.2.3.3

5.2.4.5.2 Installazione a bordo

Per l'installazione a bordo dei sensori dovrà essere previsto o un apposito contenitore dedicato ai soli sensori e connesso attraverso apposito connettore al KIT Car Bus oppure sarà direttamente integrato nel Kit.

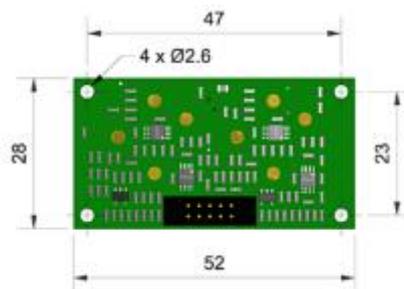
Nell'un caso e nell'altro dovranno essere previsti degli scassi che consentano ai sensori di poter entrare in contatto con l'aria in modo in modo da poter rilevare la presenza degli eventuali inquinanti.

Si riportano di seguito le dimensioni fisiche dei sensori

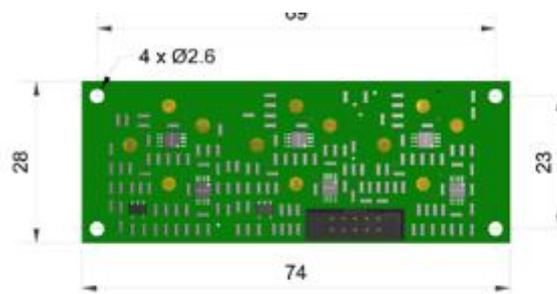


Sensore di tipologia a A4: Dimensioni Meccaniche

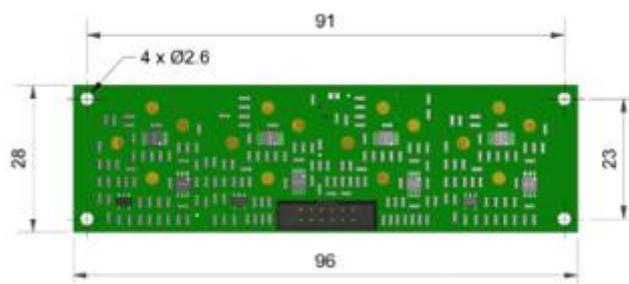
E le dimensioni delle AFE



AFE a 2 Sensori



AFE a 3 Sensori



AFE a 4 Sensori

I sensori dovranno essere installati seguendo le prescrizioni di interfacciamento meccanico descritte nel paragrafo 5.2.2.2.

5.2.5 Protocollo base Bluetooth

In questa sezione si sviluppa la specifica del protocollo di base Bluetooth, che viene indicato come *I2 – protocollo Base BT* in *Figura 25* e *Figura 29*.

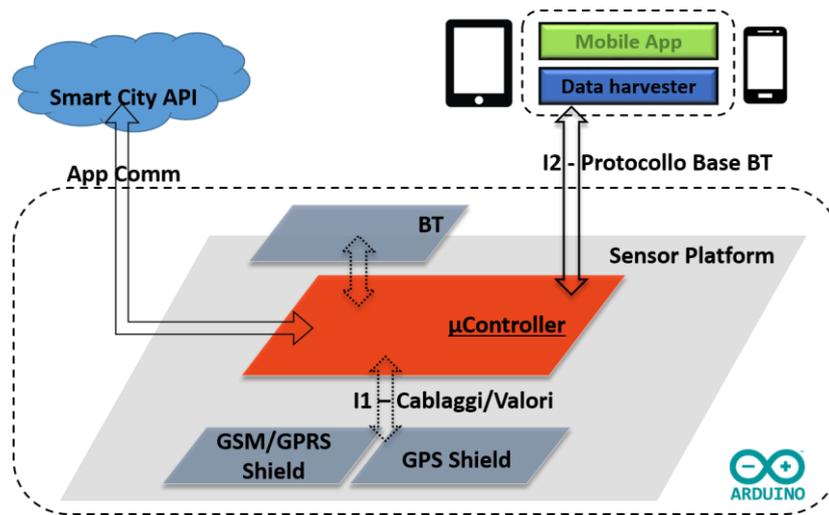
La specifica deve essere condivisa tra chi sviluppa il software della Piattaforma e del Data Harvester, in modo da utilizzare in modo consistente le funzionalità Bluetooth presenti sulla Piattaforma e sul Tablet per realizzare un meccanismo di trasporto wireless di messaggi.

In prima istanza, ci si propone di sviluppare il protocollo seguendo lo schema di comunicazione “nativo” Bluetooth Low Energy, descritto in 3.2.2.3.2. Qualora in corso lavori il throughput si rivelasse insufficiente per supportare tutte le funzioni previste, si provvederà ad un riesame generale del meccanismo per migliorarlo o per selezionarne uno alternativo.

6 Appendice A – Studio di fattibilità della funzione AATF (autonomous anti-theft feature) realizzata nel contesto del kit bike

La funzione Autonomous Anti-Theft Feature (AATF), di cui alla Sezione 3.2.3 scenario b.1.ii., si concretizza con il tracking GPS e l’instaurazione di una comunicazione via SMS e/o http con il cloud Smart City. Analogamente allo scenario b.1.i, la Piattaforma non è associata con uno Smartphone, quindi non è in grado di sfruttare il sensore GPS e la connettività di quest’ultimo. Per queste ragioni è necessario che la Piattaforma abbia un proprio sensore GPS per implementare la funzionalità di tracking della posizione, unitamente ad una shield GSM/GPRS per l’implementazione della funzionalità di connettività long range.

L’architettura di riferimento GPS + GSM/GPRS Shield è riportata nella Figura seguente:



Possibili soluzioni Hardware GSM/GPRS Shield Arduino

- Arduino GSM Shield: costo € 85,00
- Shield GSMSPRSSHIELDV2: costo € 30,00

Nel caso venga utilizzata questa soluzione implementativa della funzione AATF si ha che:

- la Piattaforma è del tutto autonoma rispetto allo Smartphone che in questo scenario non è collegato alla Piattaforma stessa;
- la notifica di uno spostamento anomalo della Piattaforma è notificato in tempo reale alla Centrale Sii-Mobility che potrà adottare le operazioni del caso.

D'altra parte, sono da tenere in considerazione i seguenti aspetti:

- sono richiesti costi aggiuntivi per l'acquisto della Shield GSM/GPRS unitamente al costo fisso di attivazione del servizio di connettività long range;
- modifica del design del VKMbike;
- implementazione della comunicazione lato client con le Smart City API;
 - valutazione delle risorse necessarie per il funzionamento delle librerie di connettività;
 - valutazione circa la necessità di cambiare board con una più performante;
- ridondanza hardware quando la Piattaforma è collegata con lo Smartphone;
- in contrasto con la scelta dello Smartphone utente che ricopre di fatto il ruolo di Gateway;

Per le motivazioni sopra esposte si decide di adottare la soluzione Autonomous Anti-Theft Feature (AATF), Sezione 3.2.3 - scenario b.1.i e Sezione 3.2.5.2. ovvero la soluzione AATF con GPS e segnale sonoro di allarme, Sezione 3.2.5.3. La soluzione AATF con GPS permette di coprire lo scenario in cui la Piattaforma è parcheggiata fuori da un'area di parcheggio attrezzata. Viceversa, se la Piattaforma è parcheggiata all'interno di un'area attrezzata, la funzione AATF con GPS può essere coadiuvata da meccanismo come tag RFID.

7 Bibliografia

- [xx] TBD