



Sii-Mobility

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

Trasporti e Mobilità Terrestre, SCN_00112

Deliverable ID: DE3.1a

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

Data corrente	17/02/2017
Versione (solo il responsabile puo' cambiare versione)	V 0-22a
Stato (draft, final)	final
Livello di accesso (solo consorzio, pubblico)	Pubblico
WP	OR3
Natura (report, report e software, report e HW..)	report
Data di consegna attesa	M7 – Luglio 2016
Data di consegna effettiva	M14 – Febbraio 2017
Referente primario, coordinatore del documento	Alessandro Paganone, ECM, alessandro.paganone@ecmre.com
Contributor	Giovanni Collodi, MIDRA, collodi@ing.unifi.it Francesco Chiti, UNIFI-LART, francesco.chiti@unifi.it Andrea Rindi, UNIFI-DIEF/Mdmlab, andrea.rindi@unifi.it

	<p>Laura Cocone, SWARCO MIZAR, laura.cocone@swarco.com Leonardo Fabbri, ELFI, leonardo.fabbri@elfisrl.net Carlo Alberto Dami, CReAI, carloalberto.dami@tin.it</p>
Coordinatore responsabile del progetto	<p>Paolo Nesi, UNIFI, paolo.nesi@unifi.it</p>

Sommario

1	Executive Summary	6
2	Introduzione ed obiettivi	6
2.1	Acronimi, sigle, terminologia.....	6
2.2	Documenti di riferimento	7
2.2.1	Standards e normative applicabili	7
2.2.2	Documenti Sii-Mobility	7
2.2.3	Altri riferimenti	7
2.3	Contesto.....	8
2.4	Notazioni	9
2.4.1	Schemi di architettura	9
2.4.2	Schemi di dettaglio.....	9
2.5	Obiettivi.....	10
3	Kit bike.....	11
3.1	Architettura generale	11
3.2	Specifica di dettaglio	12
3.2.1	Piattaforma	13
3.2.2	Funzione ATF, anti-theft feature	25
3.2.3	Smartphone utente.....	26
3.2.4	Sensori / Attuatori	27
3.2.5	Protocollo base Bluetooth	45
4	Kit carbus base	47
4.1	Architettura generale	47
4.2	Specifica di dettaglio	48
4.2.1	Interfaccia I3	49
4.2.2	Nuovo SW.....	49
4.2.3	Data Harvester, MAPP16.....	49
4.2.4	Installazione di TURMS	50
5	Kit carbus evoluto	51
5.1	Architettura generale	51
5.2	Specifica di dettaglio	52
5.2.1	Piattaforma	53
5.2.2	Tablet di bordo	61
5.2.3	Sensori.....	62
5.2.4	Protocollo base Bluetooth	68

6	Appendice A – Studio di fattibilità della funzione AATF (autonomous anti-theft feature) realizzata nel contesto del kit bike (LART)	68
---	---	----

Figura 1 :	i Kit veicolari nel contesto di Sii-Mobiliy	8
Figura 2 :	notazione grafica generale	9
Figura 3 :	notazione grafica di dettaglio	9
Figura 4 :	Architettura generale del Kit bike	11
Figura 5 :	Kit Bike - diagramma di contesto.....	12
Figura 6 :	Piattaforma – diagramma di contesto	13
Figura 7 :	Kit bike – Contenitore e disposizione apparati.....	15
Figura 8 :	Schema generale del kit bike.....	18
Figura 9 :	Vehicular Kit Manager Bike – casi d’uso	22
Figura 10:	Vehicular Kit Manager Bike – FSM	23
Figura 11 :	Smartphone – diagramma di contesto	26
Figura 12:	Breakout Board GROVE IMU 9 DOF con jumper installato per scambio dati secondo comunicazione I2C con MCU	28
Figura 13:	comunicazione I2C , interfacciamento fisico	29
Figura 14:	definizione della fase di START/STOP, ACKNOWLEDGE sul bus I2C	30
Figura 15:	comunicazione sul bus I2C (indirizzo slave + data)	30
Figura 16:	Architettura del kit di anticollisione e monitoraggio stradale	31
Figura 17:	Architettura e gestione della comunicazione I2C tra MCU e MPU-Breakout Board	31
Figura 18:	dettaglio dell’architettura interna del chip MPU 9250 (sinistra) e interfacciamento con processore esterno (destra).....	32
Figura 19 :	registro 55 dedito alla configurazione del pin INT: se INT_LEVEL è basso il livello logico del pin è attivo alto, e se INT_OPEN è basso il pin è considerato push-pull	32
Figura 20:	registro 56 dedito all’abilitazione dei possibili interrupt: settando alto il Bit0 è attivata la segnalazione tramite interrupt di dato presente su registro FIFO	32
Figura 21:	registri (59-64) che contengono i dati relativi alle misure accelerometriche per ogni asse descritto tramite 2 byte.....	32
Figura 22:	registri (67-72) che contengono i dati relativi alle misure giroscopiche per ogni asse descritto tramite 2 byte.....	33
Figura 23:	specifiche di dettaglio della scheda GROW-IMU 9250 9 DOF.....	33
Figura 24:	specifiche di dettaglio della breakout board GROV-IMU 9250 9 DOF: regolatore di tensione (a), traslazione di livello da IIC 5 V a IIC 3.3 V (b), MPU 9250 (c)	34
Figura 25:	modulo GPS VK2828U7G5LF V1.0	34

Figura 26: Scheda GPS nel contesto della architettura del Kit bike	37
Figura 27: Principio di funzionamento del sensore ultrasonico con particolare attenzione verso il posizionamento del sensore	40
Figura 28: Interfaccia fisica tra MCU (Arduino) e sensore DHT 11	40
Figura 29: Interfaccia fisica per il collegamento del sensore ultrasonico HC-SR04 e sensore di temperatura per correzione della misura della distanza	41
Figura 30: Diagramma temporale del funzionamento del sensore	42
Figura 31: Comunicazione MCU-DHT11	42
Figura 32: Logica dell'architettura del kit anti collisione.....	43
Figura 33: Architettura del sistema (KIT anticollisione)	43
Figura 34: tre sensori acustici scelti come possibile elemento sensibile ai fini del rilevamento della prossimità tra bike e veicolo che segue.....	44
Figura 35 : Architettura generale del Kit carbus base.....	47
Figura 36 : Kit carbus base - diagramma di contesto.....	48
Figura 37 : Architettura generale del Kit carbus evoluto.....	51
Figura 38 : Kit carbus evoluto - diagramma di contesto.....	52
Figura 39 : Piattaforma – diagramma di contesto	53
Figura 40 : Schema generale del kit carbus evoluto	56
Figura 41 : Carbus Kit Manager Bike – casi d'uso.....	59
Figura 42: Carbus Kit Manager Bike – FSM.....	60
Figura 43 : Smartphone – diagramma di contesto	61
Figura 44 : Connettori OBD-II – tipo “a” (sinistra) e tipo “b” (destra)	63
Figura 45 : UART Adapter per CanBus Sniffer	64

1 Executive Summary

Il documento contiene il progetto dettagliato dei kit veicolari previsti nel conteso del 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]).

Questo documento (DE3.1a) ha lo scopo di descrivere quanto si intende sviluppare. È pianificata una successiva revisione (DE3.1b) che descriverà quanto effettivamente prodotto, tenendo conto di inevitabili modifiche e variazioni in corso d’opera. Servirà anche a risolvere alcuni “TBD” che a questo stadio del progetto sono ancora presenti.

2 Introduzione ed obiettivi

2.1 Acronimi, sigle, terminologia

API	Application Program Interface
AVL - AVM	Automatic Vehicle Location - Automatic Vehicle Monitoring
CEN	European Committee for Standardization
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
PMS	Private Mobile Systems
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
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
VWSN	Vehicular Wireless Sensor Networks
W3C	World Wide Web Consortium
WSN	Wireless Sensor Networks
XMI	XML Metadata Interchange standard di OMG
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 - Specifica di Integrazione e Casi di Test
[SiiM 3]	DE8.5 - Manuale di qualità del progetto
[SiiM 4]	DE3.6a - Sensori: specifica di dettaglio dello sviluppo
[SiiM 5]	DE3.11a - 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

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 dell'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 Applicazioni Mobili (*Mobile App*) e Applicazioni per la raccolta dei dati (*Data Harvester*) fornendo loro i servizi per svolgere le loro funzioni, includendo anche le azioni descritte al punto precedente,

Mobile App e Data Harvester sono installati sui Kit veicolari scaricandoli 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.

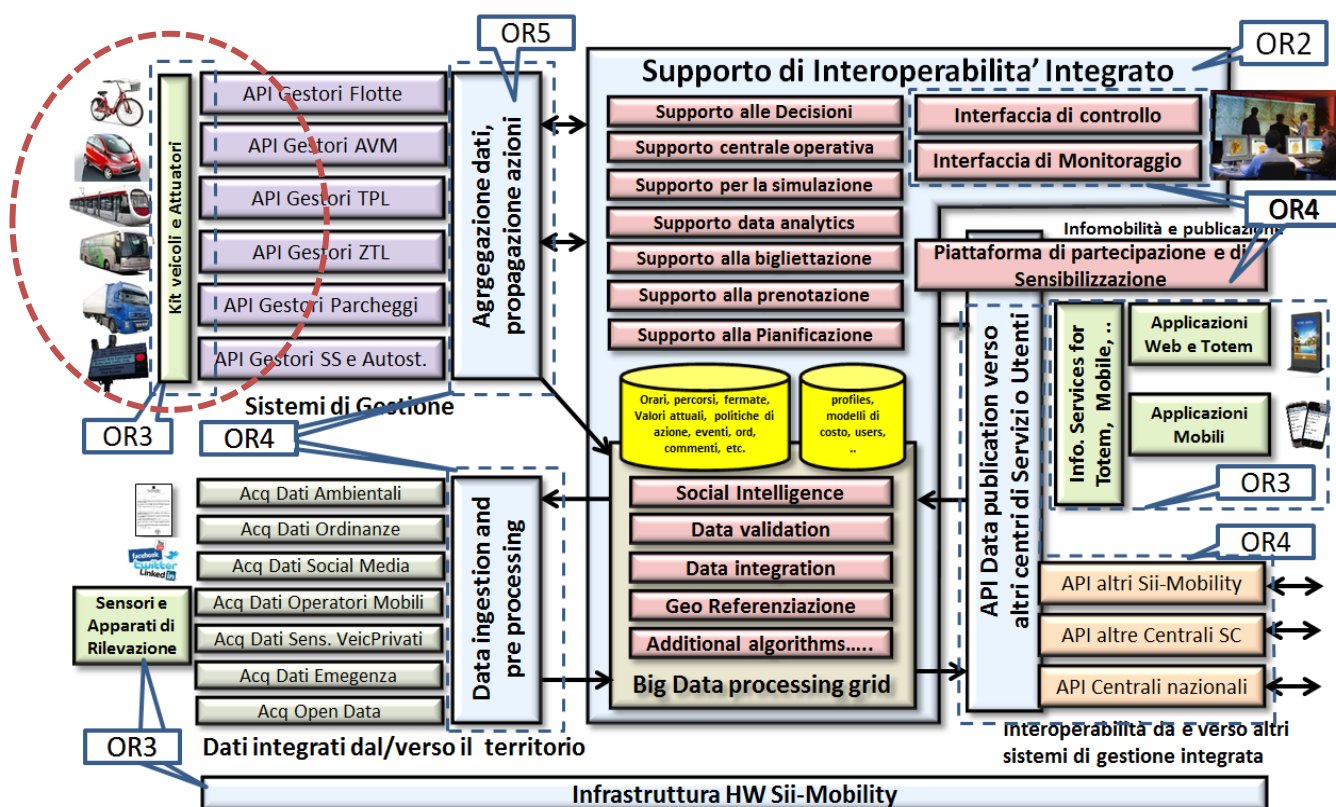


Figura 1 : i Kit veicolari nel contesto di Sii-Mobility

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.

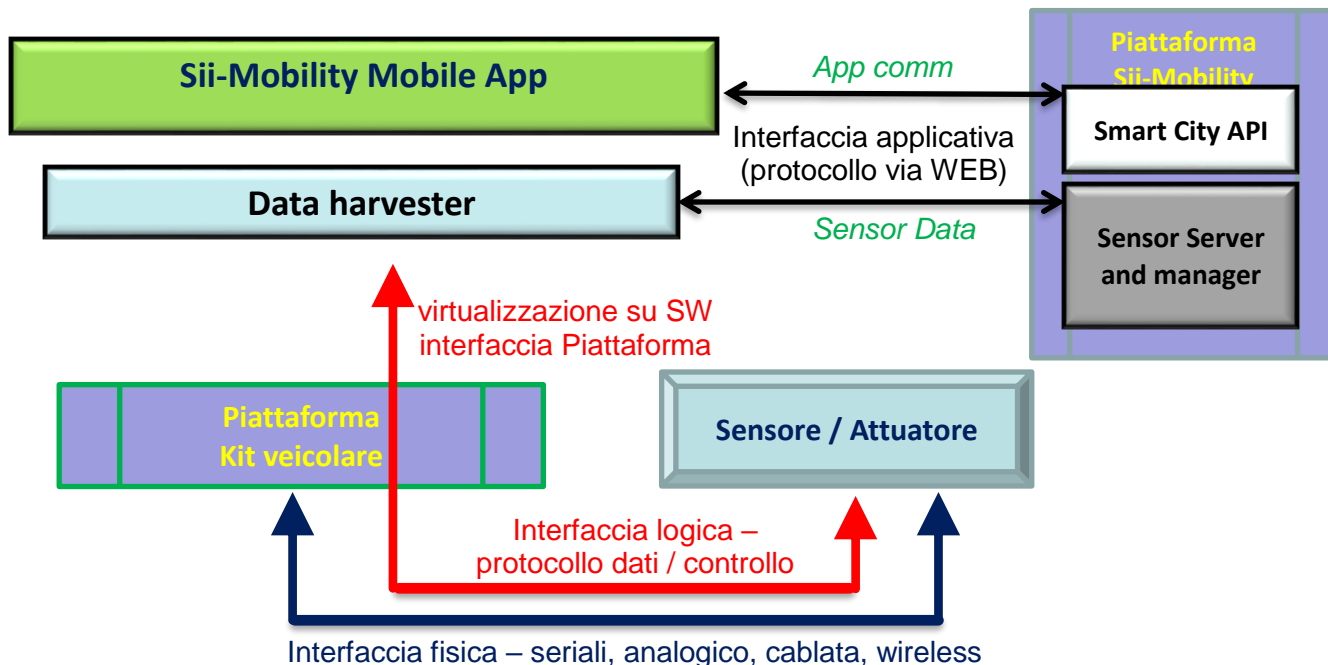


Figura 2 : notazione grafica generale

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).

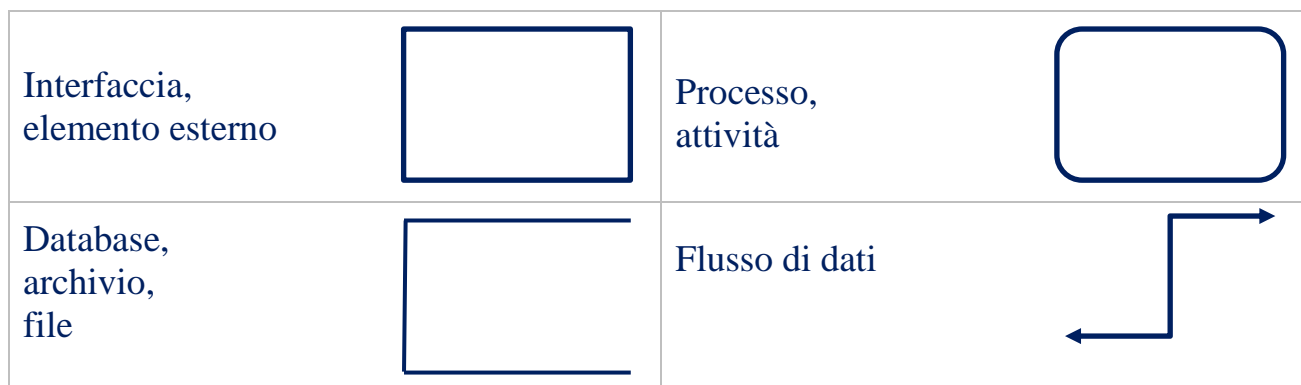





Figura 3 : notazione grafica di dettaglio

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).

2.5 Obiettivi

Questo documento intende sviluppare completamente l'architettura di dettaglio dei Kit veicolari, senza però duplicare informazioni già previsti in altri documenti di Sii-Mobility. In particolare:

-  alcuni sensori che fanno parte dei kit veicolari sono descritti in [SiiM 4], essendo prodotti della Attività 3.2. Per questi non si forniscono ulteriori informazioni ma si rimanda al documento indicato fornendo precisi riferimenti;
-  Il Data Harvester (MAPP16) nelle sue varie istanze e versioni è descritto in altri documenti che al momento non sono determinati. In questo documento sarà inserito un riferimento, al momento “TBD”, che verrà risolto nella revisione DE3.1b.
-  Le Mobile Apps (APP01, APP02, MAPP01) sono descritte in altri documenti che al momento non sono determinati. In questo documento sarà inserito un riferimento, al momento “TBD”, che verrà risolto nella revisione DE3.1b.

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 vengono scaricati il Data Harvester e la Mobile App.

La Piattaforma e lo Smartphone comunicano utilizzando gli apparati Bluetooth 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*

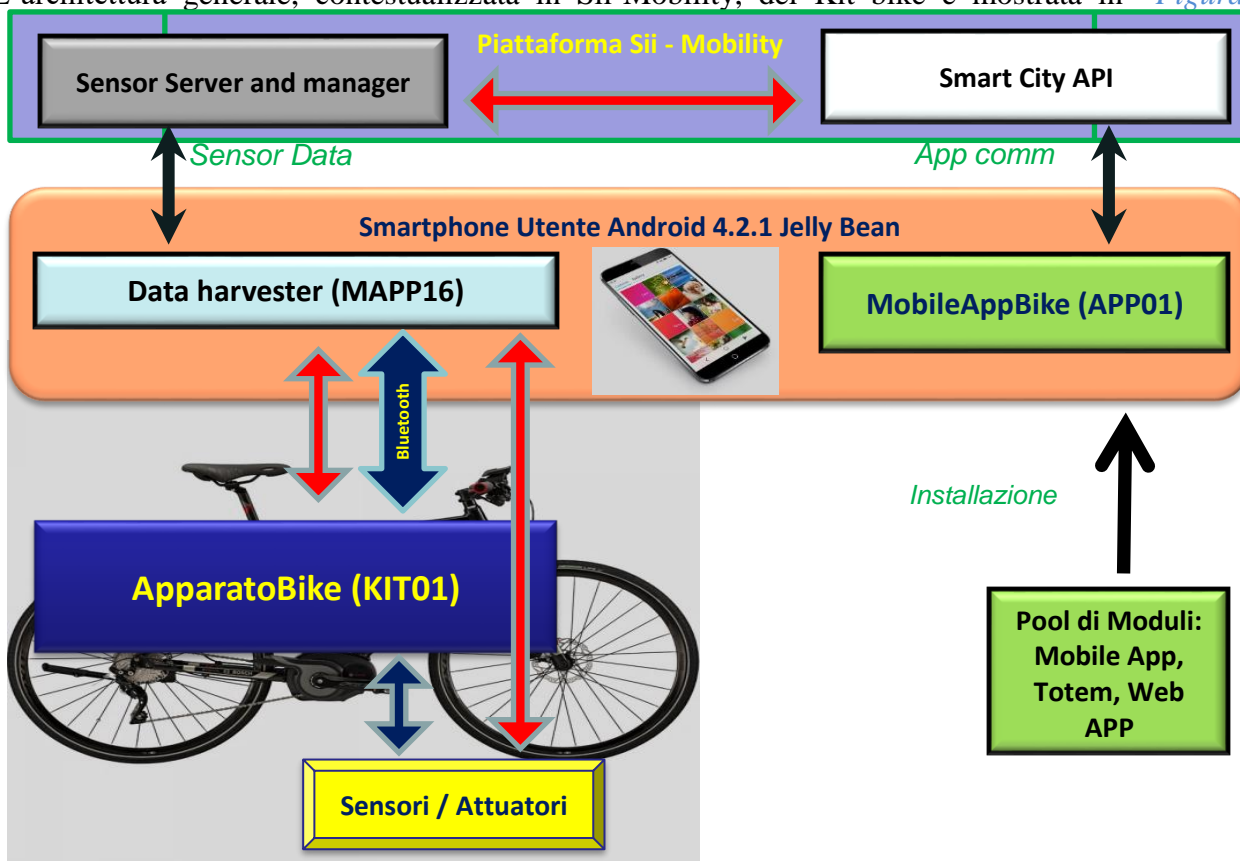


Figura 4 : Architettura generale del Kit bike

I vantaggi offerti da questa organizzazione sono molteplici e rilevanti:

- chi sviluppa Data Harvester e Mobile App 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,

- se necessario, Data Harvester e Mobile App possono comunicare direttamente tra loro e sincronizzarsi utilizzando meccanismi standard di Android, ben noti a tutti gli sviluppatori coinvolti,
- 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 3.2.2 per dettagli.

3.2 Specifica di dettaglio

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.

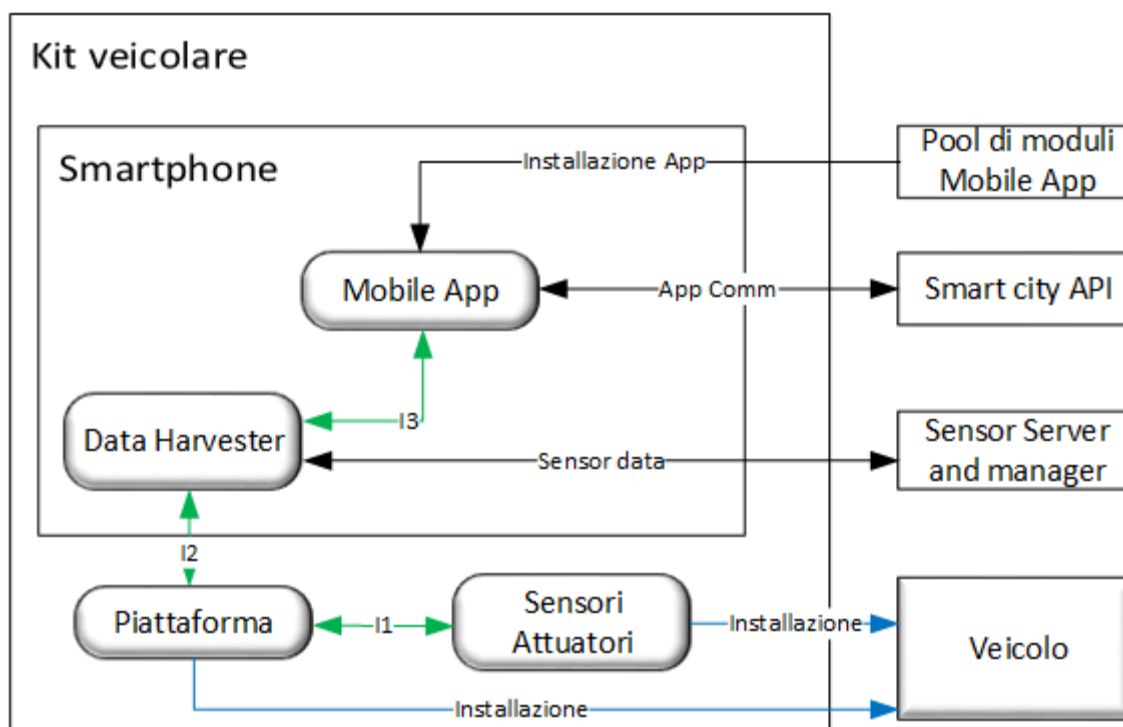


Figura 5 : Kit Bike - diagramma di contesto

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 3.2.1.4;
- 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;
 - **I3** è l'interfaccia fisica e logica tra Mobile App e Data Harvester.

3.2.1 Piattaforma

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.

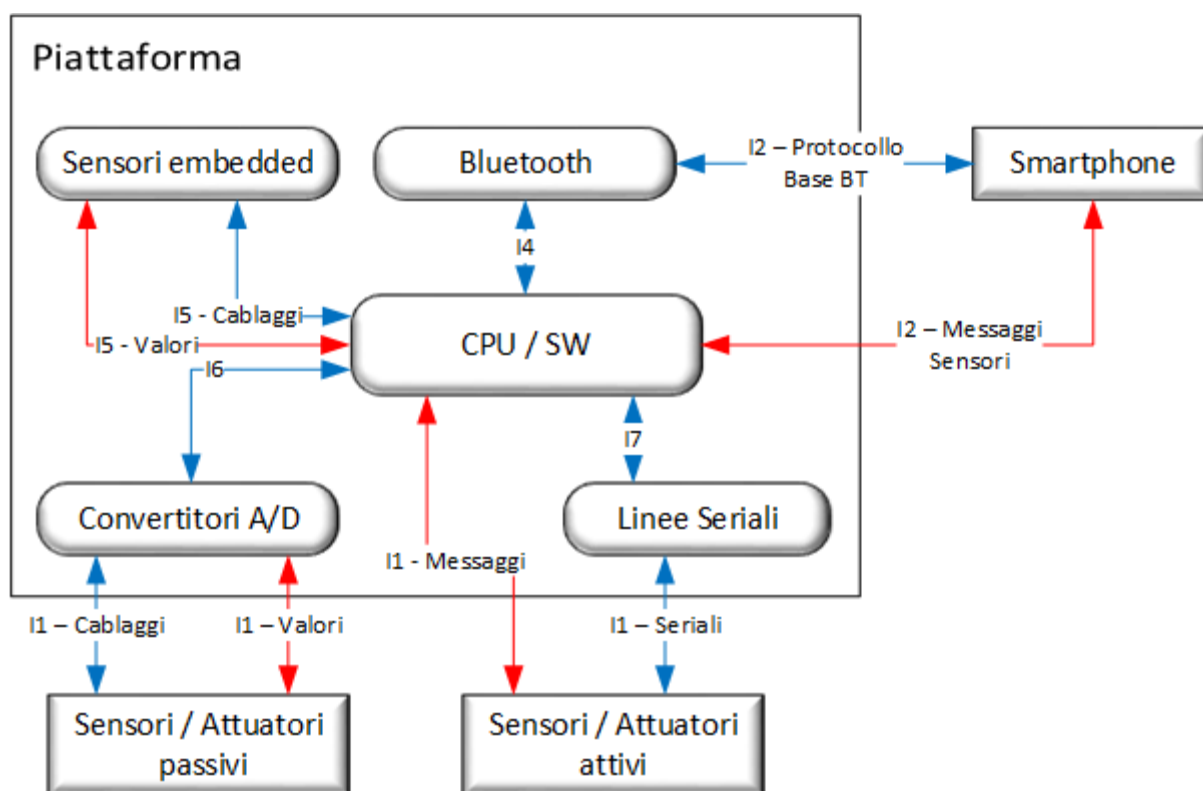


Figura 6 : Piattaforma – diagramma di contesto

La piattaforma include gli elementi seguenti, tutti descritti in dettaglio alla sezione 3.2.1.2:

- 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;
- un dispositivo Bluetooth per le comunicazioni con lo smartphone;
- una scheda CPU (motherboard).

I sensori / attuatori interfacciati sono:

- una scheda accelerometrica in grado di fornire dati per eseguire analisi sullo stato della strada e sulla condotta del veicolo. È descritta in dettaglio alla sezione 3.2.4.1
- un apparato GPS per supportare la funzione antifurto. È descritto in dettaglio alla sezione 3.2.4.2;
- un segnale sonoro di allarme per supportare la funzione antifurto e la funzione anticollisione. È descritto in dettaglio alla sezione 3.2.4.3;
- sensori ambientali, descritti in dettaglio alla sezione 3.2.4.4;
- sensori anti-collisione, descritti in dettaglio alla sezione 3.2.4.5;

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 (NEGENTIS) 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 *14*, 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 *12 – Messaggi sensori* è costituita dall'OR di:

- *11 – Valori*
- *11 - Messaggi*
- *15 – Valori*

3.2.1.1 Meccanica, contenitori, carpenterie

Con riferimento ai componenti ad oggi individuati, è stata studiata una prima soluzione per il contenitore. Questa prevede una box in alluminio anodizzato, di dimensioni compatte, che contiene tutto il kit bike, come da *Figura 7* sotto riportata.

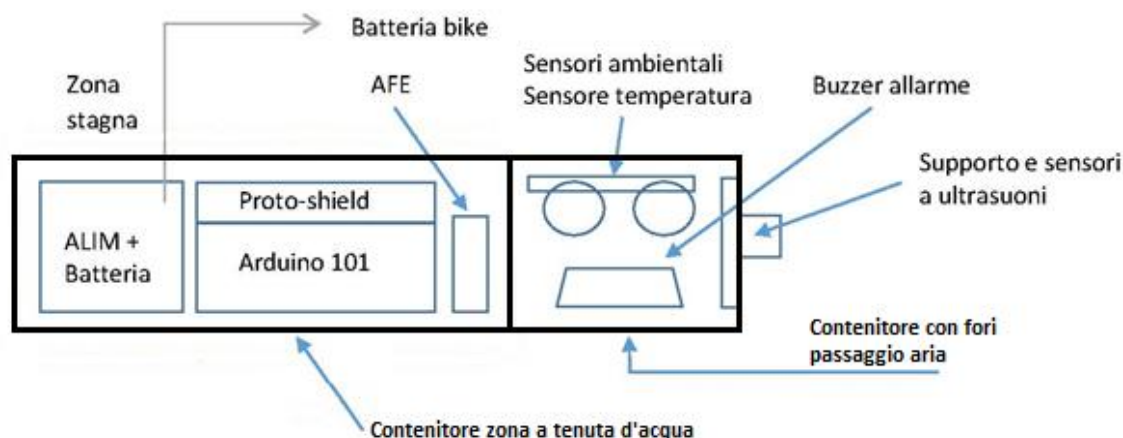


Figura 7 : Kit bike – Contenitore e disposizione apparati

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

3.2.1.2 Hardware

3.2.1.2.1 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.

Il sensore di movimento on board può considerarsi sulla carta una valida alternativa al sensore di movimento individuato (vedi sezione 3.2.4.1); questa ipotesi sarà testata e nel caso affermativo si potrà togliere un sensore esterno, alleggerendo la piattaforma sia in termini di ingombro che di costo.

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]).

3.2.1.2.2 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.

3.2.1.2.3 Interfacciamento dei sensori / attuatori

La motherboard interagisce con i sensori descritti nelle sezioni seguenti: ovvero una scheda accelerometrica (3.2.4.1), un sensore di distanza a ultrasuoni (3.2.4.5), sensori ambientali (3.2.4.4), un modulo GPS (3.2.4.2). È inoltre necessario un sensore di temperatura/umidità (3.2.4.5) 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 schedina 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 1 mostrano uno schema generale del kit e il dettaglio del collegamento delle interfacce fisiche alle porte di Arduino 101.

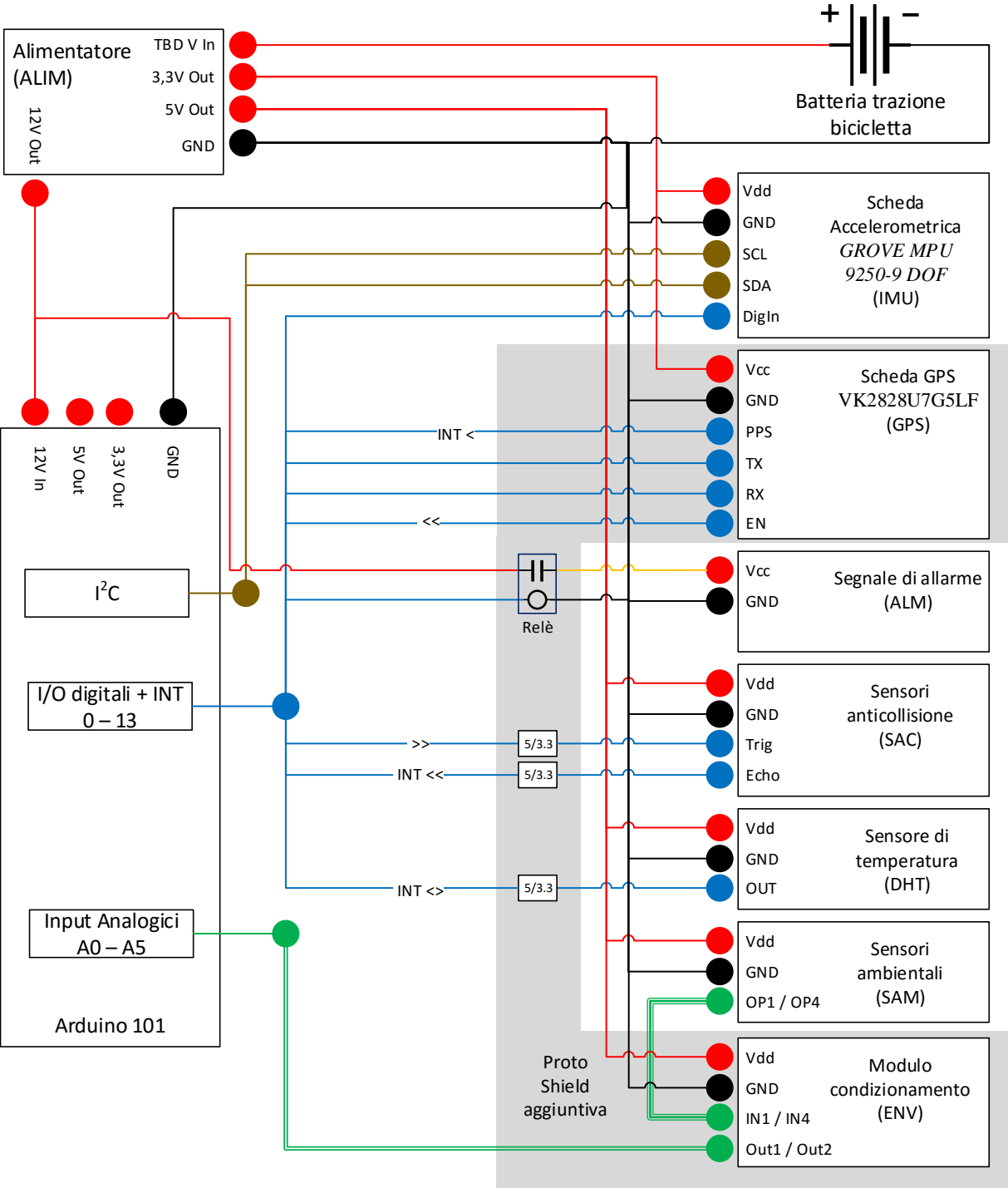


Figura 8 : Schema generale del kit bike

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		IMU DigIn
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
	IMU SCL	IMU SDA					

Tabella 1: Kit bike - connessioni fisiche tra motherboard e sensori

*: porte che supportano PWM

°: porte che supportano interrupt sui cambiamenti di stato

3.2.1.3 Software

In questo paragrafo è descritto il design del firmware della Piattaforma, denominato Vehicular Kit Manager Bike (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 3.2.5 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 3.2.2.

3.2.1.3.1 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. Indipendentemente 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 3.2.5 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 3.2.5 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 3.2.5 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 3.2.5.2 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 3.2.5 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

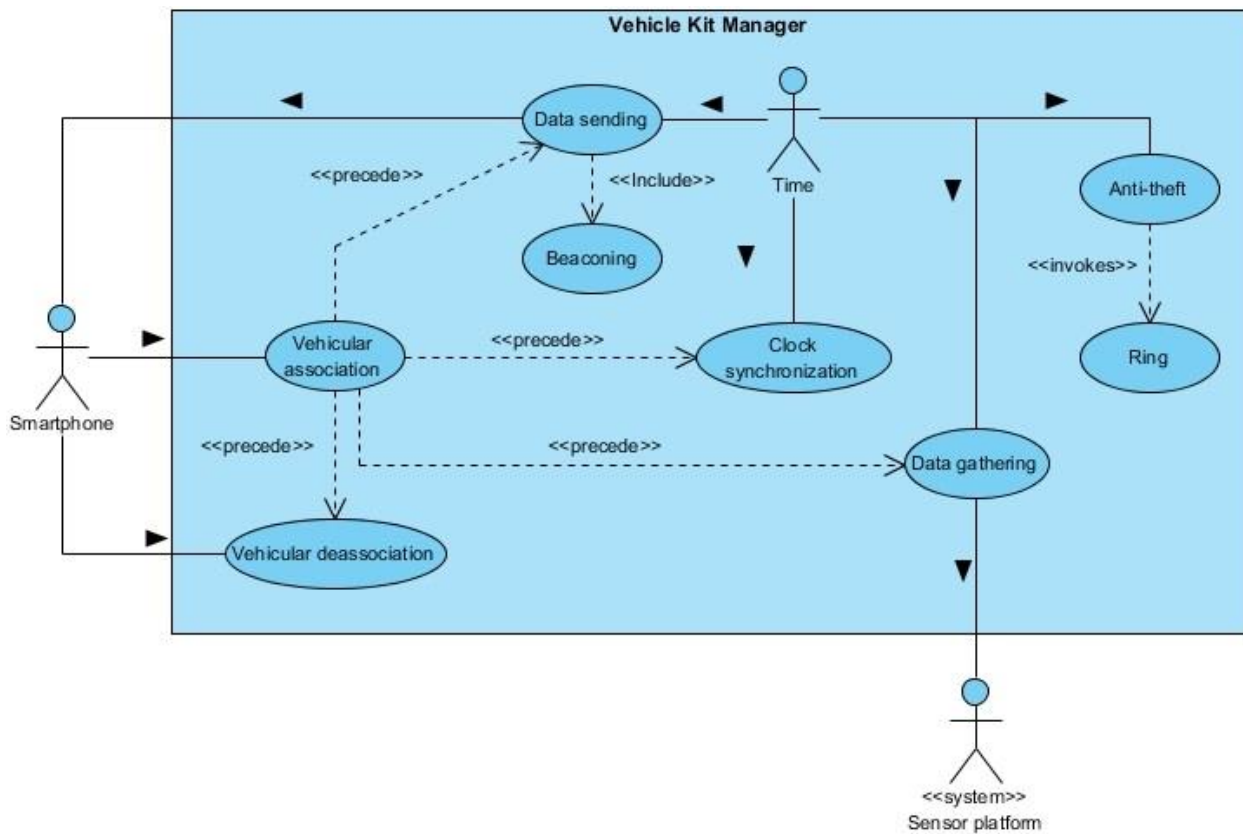


Figura 9 : Vehicular Kit Manager Bike – casi d’uso

3.2.1.3.2 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.

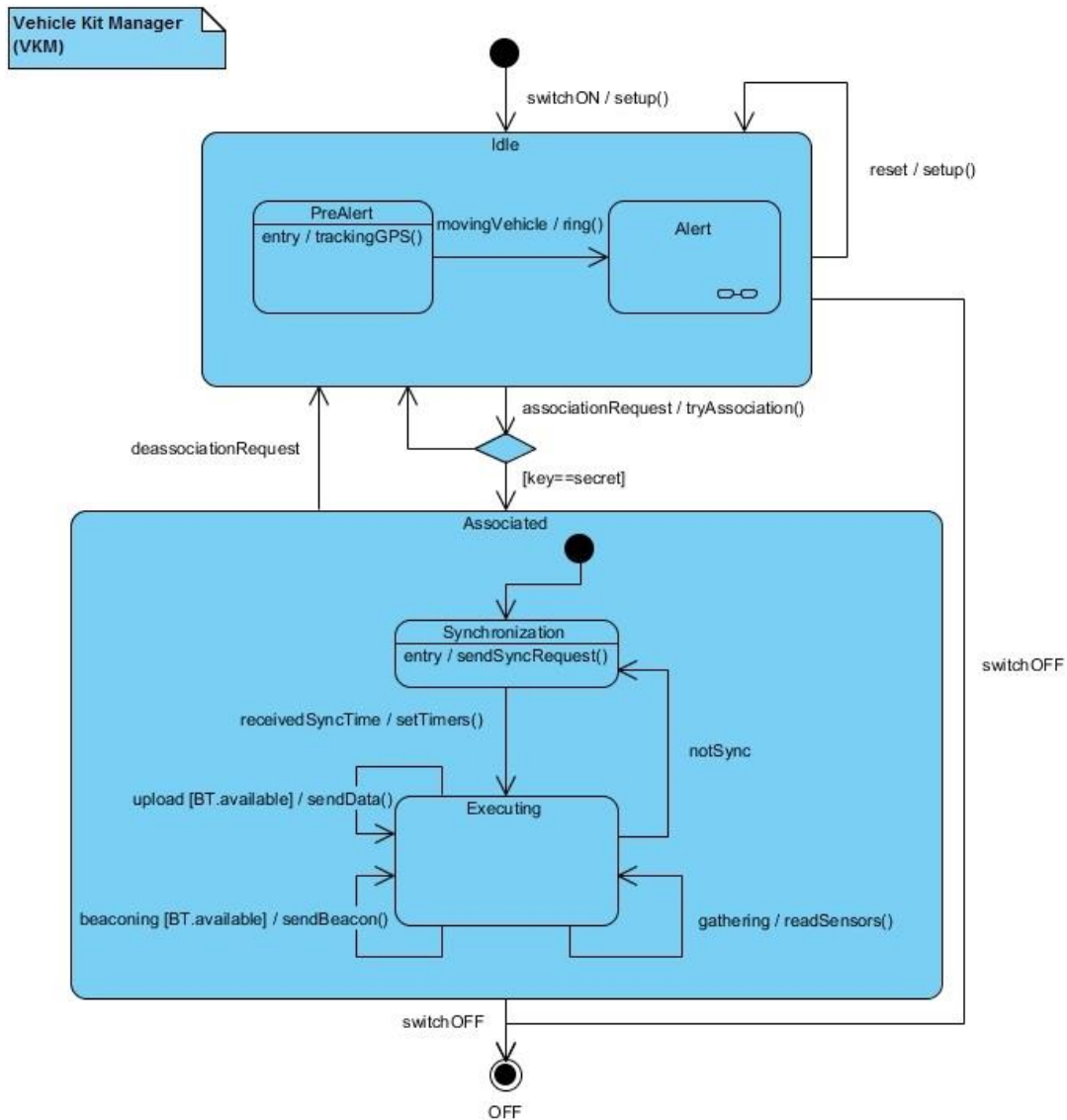


Figura 10: Vehicular Kit Manager Bike – FSM

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.

3.2.1.3.3 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 3.2.1.3.2 per realizzare i casi d'uso descritti in 3.2.1.3.1. 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 3.2.5 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** - 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.

3.2.1.4 *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.

In prima istanza si ritiene che il contenitore descritto alla sezione 3.2.1.1 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.

Una volta reperite le informazioni necessarie, le modalità di installazione a bordo e relativi supporti e carpenterie saranno descritti nella revisione DE3.1b di questo documento a cura di CReAI.

3.2.2 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 3.2.3.3.
- 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 3.2.4.2. e un Segnale sonoro di allarme, descritto alla sezione 3.2.4.3. 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. Si tenga presente che la Piattaforma non ha un sistema operativo a bordo. 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*. Da verificare se è possibile caricare sulla Piattaforma un browser disponibile open-source. Da verificare se la comunicazione può essere realizzata mediante SMS a un numero predefinito.

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.

3.2.3 Smartphone utente

La [Figura 11](#) mostra il diagramma di contesto dello Smartphone Utente

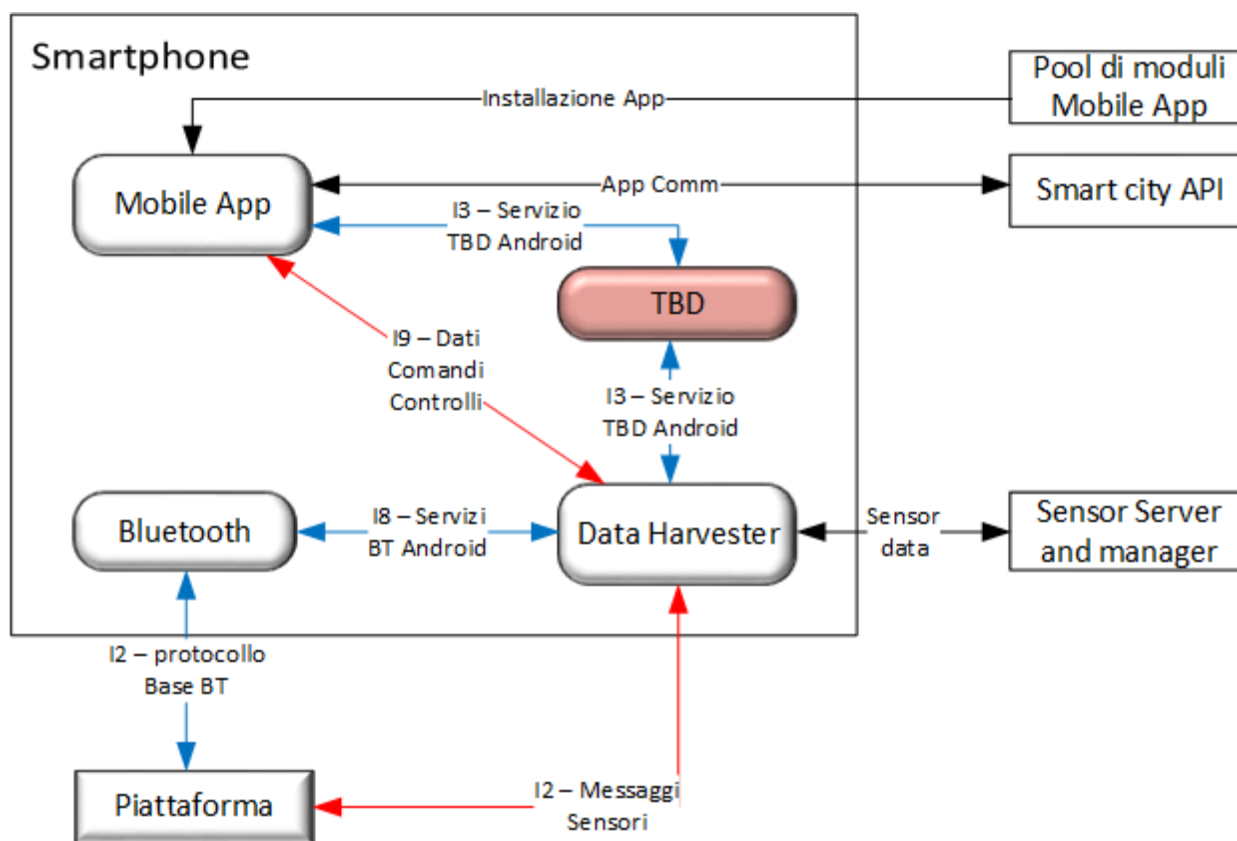


Figura 11 : Smartphone – diagramma di contesto

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

3.2.3.1 Data Harvester, MAPP16

Il Data Harvester (MAPP16) nelle sue varie istanze e versioni è descritto in altri documenti che al momento non sono determinati. Il riferimento, attualmente “TBD”, verrà risolto nella revisione DE3.1b di questo documento a cura di NEGENTIS.

3.2.3.2 MobileAppBike, APP01

La Mobile App APP01 è descritta in altro documento di progetto, che al momento non è determinato. Il riferimento, attualmente “TBD”, verrà risolto nella revisione DE3.1b di questo documento a cura di DISIT

3.2.3.3 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.

3.2.4 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.

3.2.4.1 Scheda accelerometrica

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 da una unità di misura inerziale (IMU), la quale mette a disposizione un accelerometro e giroscopio triassiali le cui informazioni sono usate per estrarre l’informazione dell’accelerazione verticale, indicatore delle condizioni del manto stradale.

La scheda deve garantire buone prestazioni dei sensori integrati, mantenendo allo stesso tempo i costi contenuti in vista di una produzione in serie del kit: di fronte a queste esigenze è stato scelto il modulo MPU 9250 che integra anche una bussola e sensore di temperatura del chip, fondamentale per la correzione delle derive termiche dei sensori principali (accelerometro, giroscopio). In aggiunta, è presente un DMP (*Digital Motion Processor*) che acquisisce e processa i dati provenienti dai sensori, alleggerendo così anche il carico computazionale che sarebbe a carico della piattaforma. La Tabella 1 riassume le caratteristiche citate.

MPU 9250	
Alimentazione	(2.375-3.46) V
Interfacce	<ul style="list-style-type: none"> • SPI (1 MHz) • I^2C (400 kHz)
Sensori	<ul style="list-style-type: none"> • Giroscopio • Accelerometro • Bussola • Temperatura
Output Rate (Acc,Gyro)	1 kHz
ADC	16 bit
Caratteristiche peculiari	<ul style="list-style-type: none"> • DMP integrato • Condizionamento del segnale • Compensazione errore di bias • Fondo scala sensori programmabile

Tabella 2: Caratteristiche principali della IMU 9250 selezionata

3.2.4.1.1 Interfaccia fisica

La scheda ospitante la IMU oggetto di studio è la *Breakout Board GROVE MPU 9250-9 DOF*, sul quale c'è anche un jumper per l'interfacciamento con il μC , parte integrante della piattaforma bike. Per la gestione della comunicazione MPU 9250 fornisce due protocolli di comunicazione (come indicato in Tabella 1):

- **SPI**: gestisce la comunicazione con accelerometro/giroscopio triassiale;
- **I^2C** : gestisce la comunicazione con accelerometro/giroscopio/magnetometro triassiale.

Entrambi i protocolli di comunicazione presentano vantaggi/svantaggi: per l'applicazione in oggetto è stato scelto il protocollo di comunicazione I^2C .

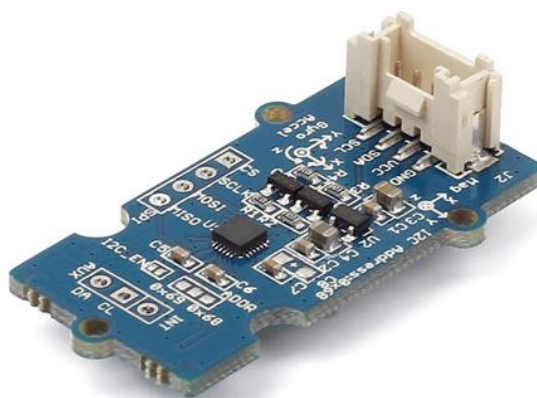


Figura 12: Breakout Board GROVE IMU 9 DOF con jumper installato per scambio dati secondo comunicazione I^2C con MCU

I collegamenti fisici da implementare hanno lo scopo di garantire l'alimentazione verso i sensori integrati e la comunicazione seriale secondo protocollo I^2C .

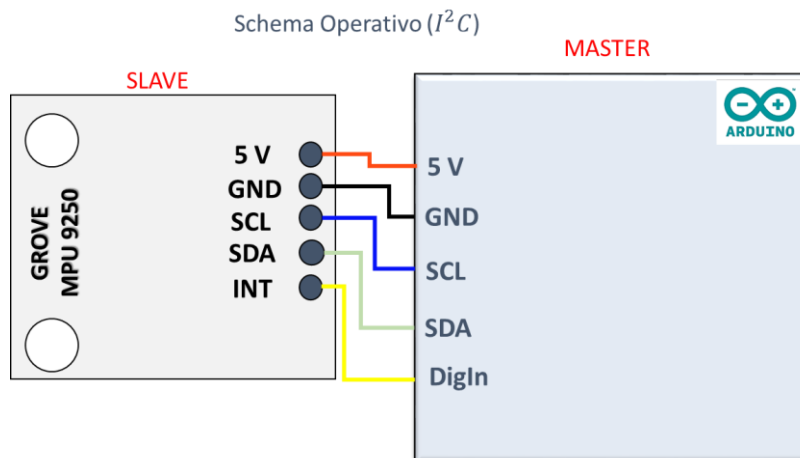


Figura 13: comunicazione I^2C , interfacciamento fisico

Figura 13 mette in luce i collegamenti fisici per implementare la comunicazione I^2C : in particolare la breakout board tollera l'alimentazione esterna a 5 V, grazie ad un regolatore di tensione interno che trasla il livello da 5 V a 3.3 V. Inoltre, il modulo ha integrato un traslatore di livello per gestire la comunicazione I^2C da 5 V a 3.3 V (vedi 3.2.4.1.4). Questo livello di tensione sarà il livello logico di tutti i pin del modulo e anche la tensione di alimentazione del chip MPU 9250. Oltre al riferimento di massa, ci sono le linee di scambio dati e clock della comunicazione I^2C e la linea di interrupt programmabile per la gestione della comunicazione stessa (vedi 3.2.4.1.3).

3.2.4.1.2 Interfaccia logica

Le linee fondamentali previste dal protocollo I^2C sono:

- SDA: Linea di scambio dati
- SCL: Linea di clock

Figura 14 mette in luce le condizioni operative che permettono di definire la fase iniziale e finale della comunicazione:

- **START**: avviene per una transizione alto-basso di SDA quando SCL è alto;
- **STOP**: avviene con una transizione basso-alto di SDA quando SCL è basso.

Avvenuta la condizione di **START**, è possibile iniziare la comunicazione, per cui ogni byte trasmesso deve essere seguito dal messaggio di **ACK**: il clock per il segnale di **ACK** viene generato dal **MASTER**, mentre lo **SLAVE** genera il messaggio di **ACK** portando bassa la linea SDA per tutto il tempo in cui il **CLOCK** è a livello alto.

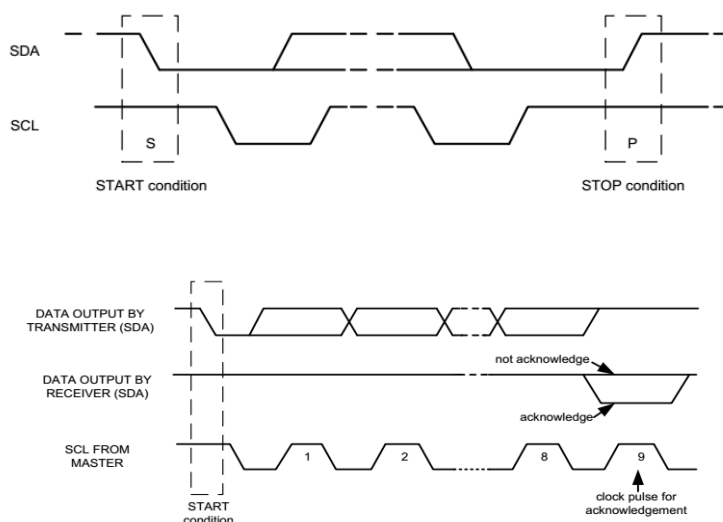


Figura 14: definizione della fase di START/STOP, ACKNOWLEDGE sul bus I^2C

La comunicazione quindi si compone delle seguenti fasi:

- Trasmissione dal MASTER di 7 bit di indirizzo dello SLAVE + un ottavo bit (R/\bar{W}) in cui si specifica se il MASTER è in lettura o scrittura (ai fini dell'applicazione l'ottavo bit sarà posto alto, imponendo il master in lettura dei dati della MPU 9250);
- Il master attende il messaggio di ACK da slave che viene segnalato secondo le condizioni descritte in Figura 2;
- La comunicazione si interrompe una volta imposte le condizioni di STOP descritte in Figura 14 e Figura 15.

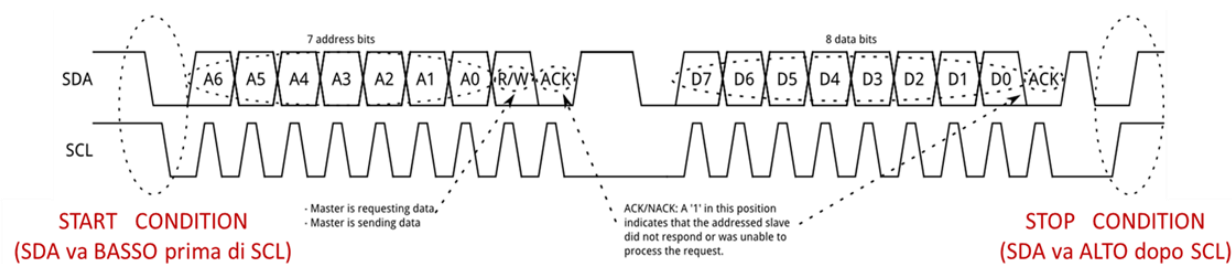


Figura 15: comunicazione sul bus I^2C (indirizzo slave + data)

3.2.4.1.3 Architettura

La scheda IMU dovrà essere montata come parte dell'unità posteriore posizionata sul portapacchi della bicicletta, parallelamente al piano del portapacchi (vedi Figura 16).

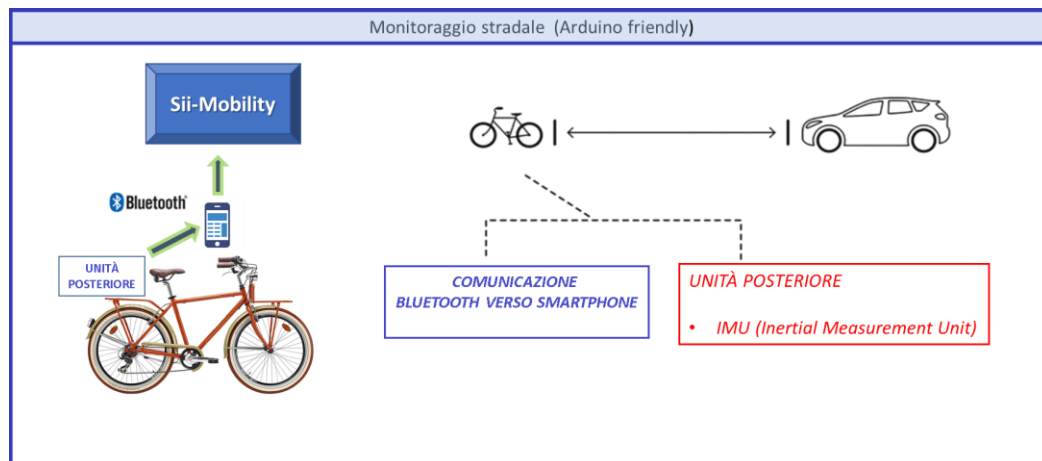


Figura 16: Architettura del kit di anticollisione e monitoraggio stradale

Figura 17 schematizza l'implementazione della comunicazione I^2C tra MASTER (μC) e SLAVE (Breakout Board GROVE MPU 9250). Dato che i pin di uscita delle comunicazione seriale sono di tipo open-drive, c'è bisogno della resistenze di pull-up che innalzano la tensione al livello logico alto (uguale alla tensione di alimentazione dei pin digitali I/O V_{DDIO} e del modulo MPU 9250 V_{DD} , vedi 3.2.4.1.4). Come detto in precedenza, la breakout board tollera una tensione di alimentazione V_{DD} a 5V grazie al suo regolatore di tensione interno che la trasla ad un livello di tensione di 3.3 V, compatibile con il range di tensione di alimentazione del chip MPU 9250 (vedi 3.2.4.1.4). Oltre le linee principali di comunicazione è inserita la linea di interrupt programmabile, che in tal caso segnerà al μC la presenza di un dato nel registro FIFO del MPU 9250.

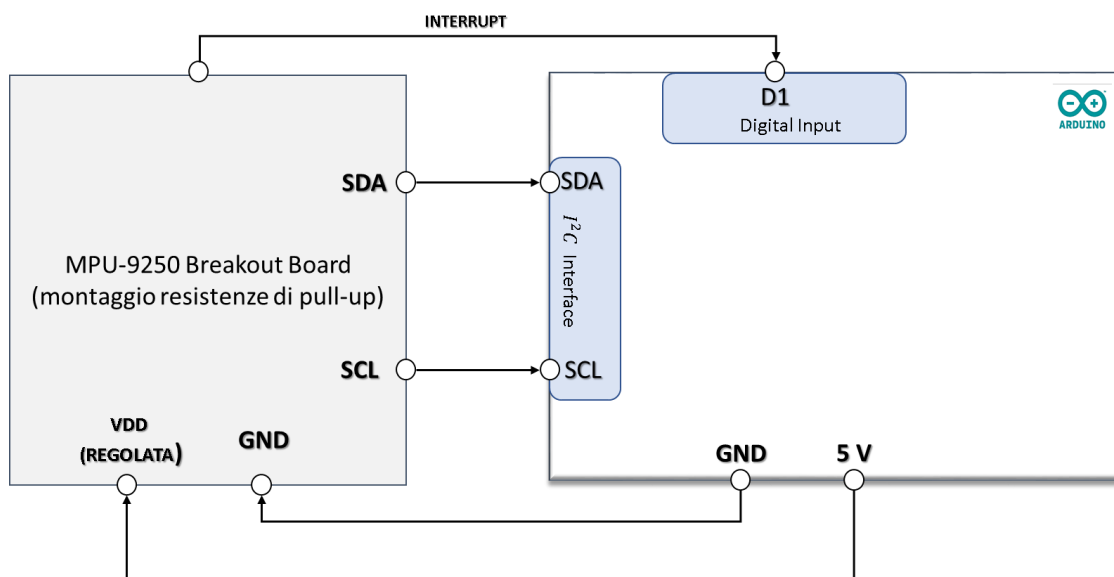


Figura 17: Architettura e gestione della comunicazione I^2C tra MCU e MPU-Breakout Board

3.2.4.1.4 Specifica di dettaglio

Figura 18 mette in evidenza l'architettura della MPU 9250 con il condizionamento dei segnali provenienti da giroscopi e accelerometri triassiale e la trattazione del magnetometro come sensore ausiliario. Le linee di uscita SCL e SDA necessitano di due resistenze di pull-up (5-10 kΩ).

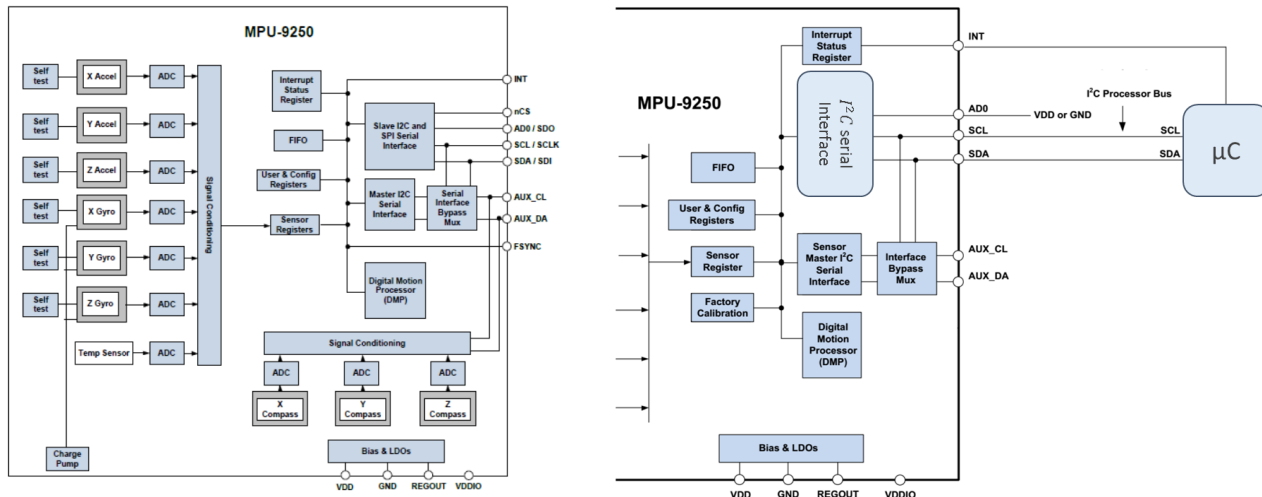


Figura 18: dettaglio dell'architettura interna del chip MPU 9250 (sinistra) e interfacciamento con processore esterno (destra)

Di seguito sono indicati i registri sui cui scrivere per settare il livello logico del pinINT (vedi Figura 19), il tipo di sorgente che innesca l'interrupt verso il μC (Figura 20) e i registri dati dell'accelerometro e giroscopio triassiali (Figura 21, Figura 22).

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
37	55	INT_LEVEL	INT_OPEN	LATCH_INT_EN	INT_RD_CLEAR	FSYNC_INT_LEVEL	FSYNC_INT_EN	I2C_BYPASS_EN	-

Figura 19 : registro 55 dedicato alla configurazione del pin INT: se INT_LEVEL è basso il livello logico del pin è attivo alto, e se INT_OPEN è basso il pin è considerato push-pull

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
38	56		MOT_EN		FIFO_OFLOW_EN	I2C_MST_INT_EN	-	-	DATA_RDY_EN

Figura 20: registro 56 dedicato all'abilitazione dei possibili interrupt: settando alto il Bit0 è attivata la segnalazione tramite interrupt di dato presente su registro FIFO

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT[7:0]							

Figura 21: registri (59-64) che contengono i dati relativi alle misure accelerometriche per ogni asse descritto tramite 2 byte

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
43	67	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT[7:0]							

Figura 22: registri (67-72) che contengono i dati relativi alle misure giroscopiche per ogni asse descritto tramite 2 byte

Figura 23 indica la *Breakout Board GROVE MPU 9250* con descrizione delle componenti principali, tra cui il jumper per il collegamento con il μ C, la selezione dell'indirizzo della scheda nella comunicazione I^2C , uscita dell'interrupt programmabile.

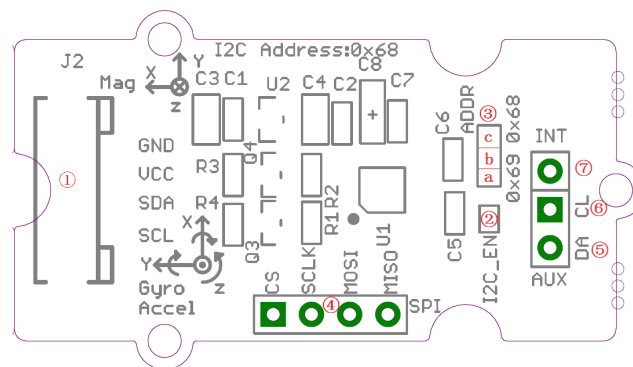


Figura 23: specifica di dettaglio della scheda GROW-IMU 9250 9 DOF

- ① Jumper per interfaccia I^2C
- ② Selezione della comunicazione: I^2C (default)
- ③ I2C Slave Address LSB (AD0): Low: 0b1101000 \Rightarrow 0x68 (default), High: 0b1101001 \Rightarrow 0x69
- ④ SPI Interfaccia
- ⑤ Master serial data per sensore ausiliario (I^2C)
- ⑥ Clock per sensore ausiliario (I^2C)
- ⑦ Uscita digitale dell'interrupt programmabile

La breakout board implementa un regolatore di tensione da V_{cc} a $V_{cc_3.3}$ V (vedi Figura 24a), un traslatore di livello con cui è possibile gestire l'interfacciamento tra comunicazioni I^2C a 5V con quelle a I^2C a 3.3 V e resistenze di pull-up a 10 k Ω (vedi Figura 24b).

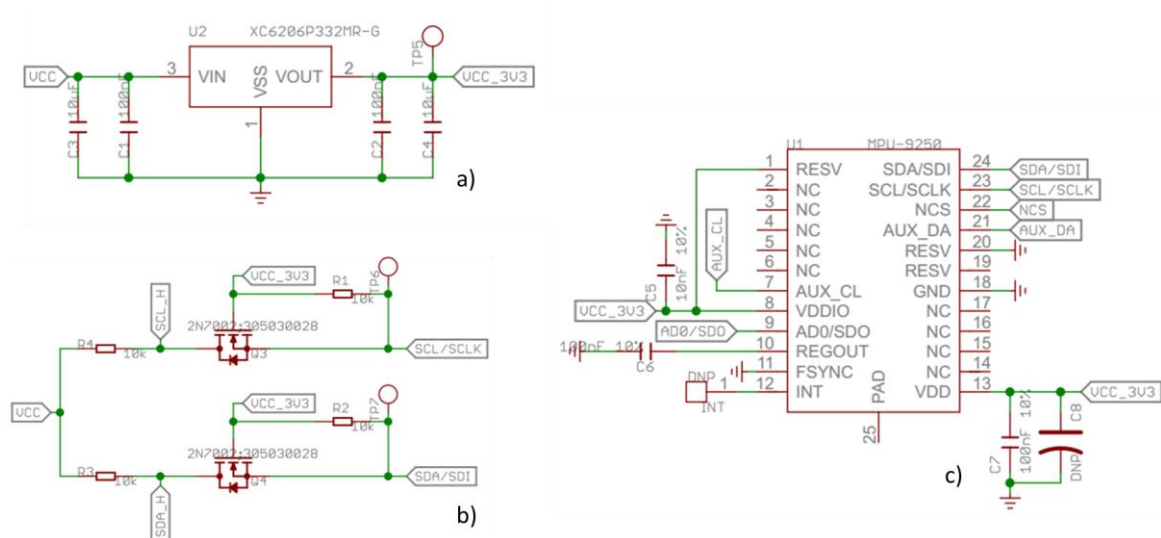


Figura 24: specifica di dettaglio della breakout board GROV-IMU 9250 9 DOF: regolatore di tensione (a), traslazione di livello da IIC 5 V a IIC 3.3 V (b), MPU 9250 (c)

3.2.4.2 Sensore GPS antifurto

La funzione Autonomous Anti-Theft Feature (AATF), Sezione 3.2.2 scenario b.1.i., si concretizza con il tracking della posizione GPS, (caso d'uso UC6 alla Sezione 3.2.1.3, 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 25).

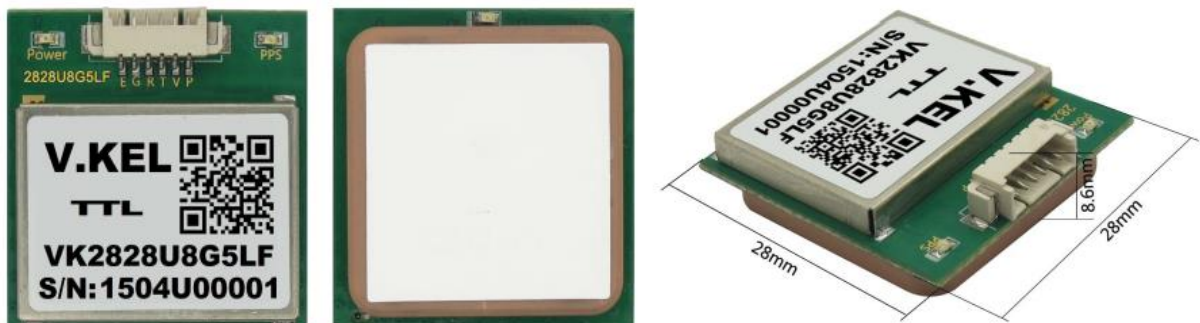


Figura 25: modulo GPS VK2828U7G5LF V1.0

3.2.4.2.1 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.

3.2.4.2.2 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

3.2.4.2.3 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.

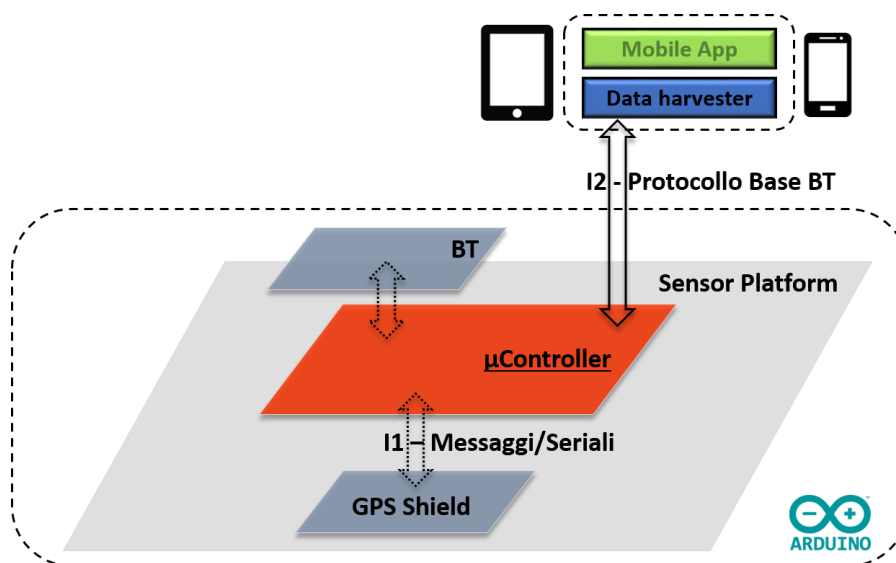


Figura 26: Scheda GPS nel contesto della architettura del Kit bike

3.2.4.2.4 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.

3.2.4.3 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 3.2.1.2.3.

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.

3.2.4.3.1 Interfaccia fisica e logica

È costituita da una singola porta di I/O di Arduino 101, identificata in Tabella 1

3.2.4.3.2 Architettura

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

3.2.4.4 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 3.2.1.1 del documento.

3.2.4.5 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 4-6 m) sia requisiti di costo e dimensioni ridotte. In prima analisi è stato scelto di usare un sensore ultrasonico con due elementi separati, Tx e Rx di cui gli aspetti da considerare sono la massima distanza rilevabile (vedi Figura 27a) e il posizionamento del sensore (vedi Figura 27b).

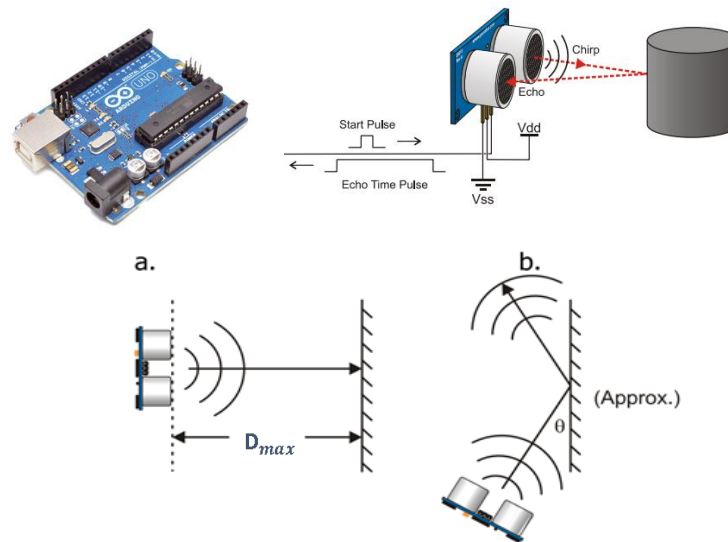


Figura 27: Principio di funzionamento del sensore ultrasonico con particolare attenzione verso il posizionamento del sensore

Il principio di funzionamento del sensore si basa sul tempo di volo dell'onda trasmessa a seguito di una eccitazione (*StartPulse*, vedi Figura 1): 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.

3.2.4.5.1 Interfaccia fisica

La descrizione delle interfacce fisiche (che in tal caso sono collegamenti elettrici) dipende dalla definizione dei pin di In/Out del sensore ultrasonico e di temperatura, i cui modelli selezionati (vedi 3.2.4.5.4) sono:

- **HC-SR04**
- **DHT-11** (temperatura e umidità)

Per entrambi i sensori ci saranno i pin dedicati all'alimentazione e GND ed eccitazione dell'onda sonora, ascolto dell'eco per il sensore ad ultrasuoni mentre la lettura dei dati per il sensore di temperatura (vedi Figura 28): è opportuno aggiungere una resistenza di pull-up sul pin di lettura dati del DHT11.

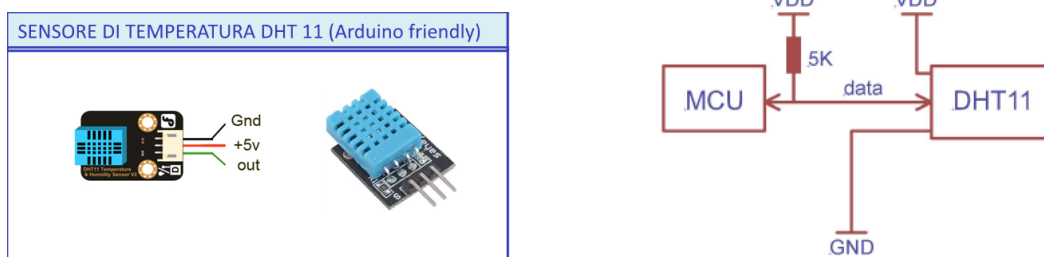


Figura 28: Interfaccia fisica tra MCU (Arduino) e sensore DHT 11

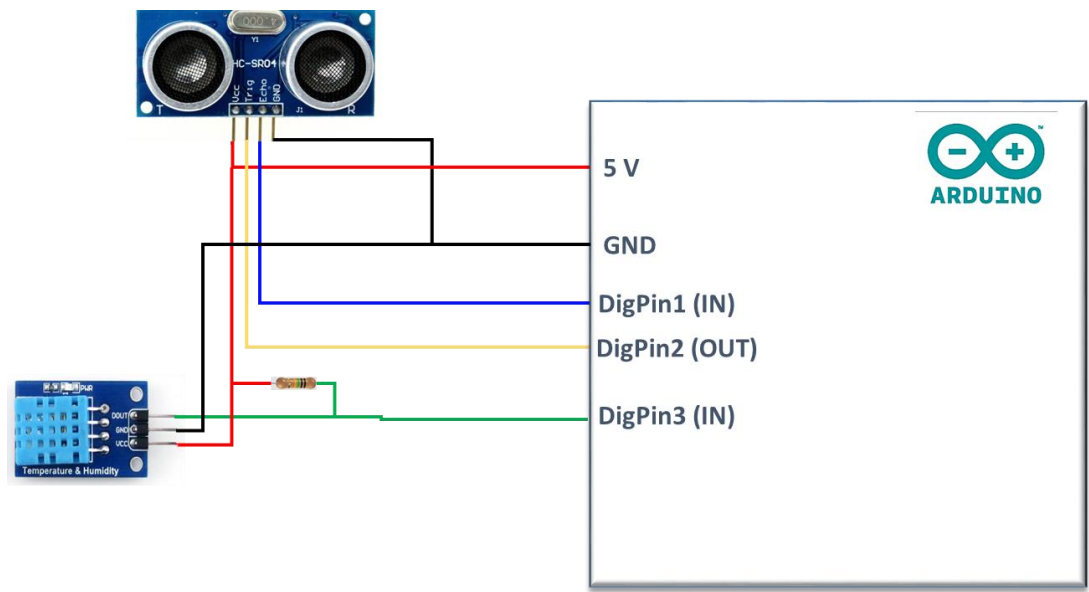


Figura 29: Interfaccia fisica per il collegamento del sensore ultrasonico HC-SR04 e sensore di temperatura per correzione della misura della distanza

3.2.4.5.2 Interfaccia logica

Per quanto riguarda le interfacce logiche, sia le uscite del sensore HC-SR04 che DHT-11 sono TTL (5 V), compatibili con l’hardware analizzato sul mercato (Arduino Like, vedi 3.2.4.5.4).

In particolare, per quanto riguarda il sensore ultrasonico si ricordano i collegamenti con la scheda (vedi Tabella 6) e i segnali logici scambiati che sono un input (livello logico TTL) di 10 μS per eccitare il sensore il quale a seguito trasmette un burst ultrasonico che porta alla ricezione dell’echo, riflesso dal target.

Pin HC-SR04	Descrizione	Link on Board
Vcc	Tensione di alimentazione	5V
Echo	Pin di ascolto dell’eco ultrasonico	Pin di ingresso Board
Trig	Pin di innesco del fascio ultrasonico in trasmissione	Pin di uscita Board
GND	Pin GND del sensore	GND

Tabella 6: Collegamenti sensore HC-SR04 con piattaforma Arduino

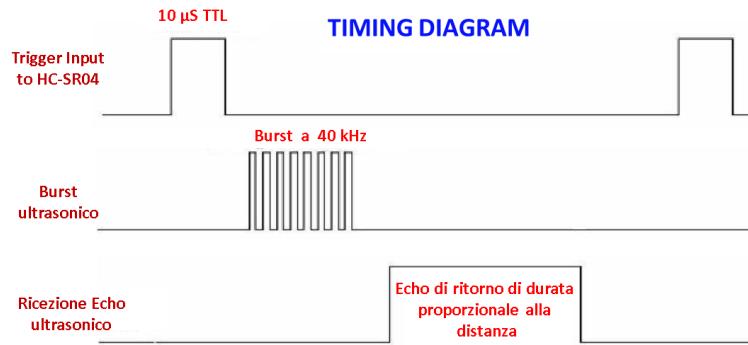


Figura 30: Diagramma temporale del funzionamento del sensore

Per quanto riguarda il DHT-11, la comunicazione con il MCU avviene nel seguente modo:

- 1 s di attesa dall'accensione, a seguito il sensore è in stand by in attesa ricevere lo START SIGNAL (livello basso di almeno 18 ms, seguito da un livello alto per almeno 40µs);
- dopo il riconoscimento dello START SIGNAL, la linea dati viene posta a livello basso per 80µs per poi essere riportata a livello alto per altri 80µs, a seguito comincia la trasmissione;
- ogni bit trasmesso inizia con un livello basso di 50µs per poi passare a livello alto, la cui durata dipende dal valore del bit (27 µs se alto, 70 µs se basso);
- al termine della comunicazione il DHT11 pone la linea dati a livello basso per 50µs per portarla a seguito a livello alto e tornare in modalità standby.
- in totale sono inviati 40 bit (ciclo di comunicazione di 40 ms) secondo la sequenza:

DATA: 8 bit per parte intera dell'RH + 8 bit per la parte decimale dell'RH + 8 bit per la parte intera T + 8 bit per la parte decimale T + 8 bit per il checksum.

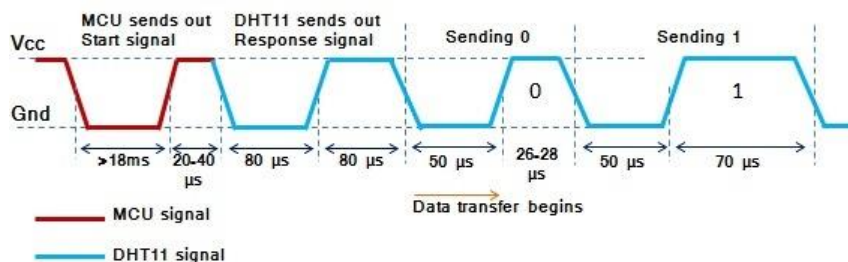


Figura 31: Comunicazione MCU-DHT11

3.2.4.5.3 Architettura

L'architettura generale all'interno della quale si colloca il sensore di prossimità è descritta in Figura 32: 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.

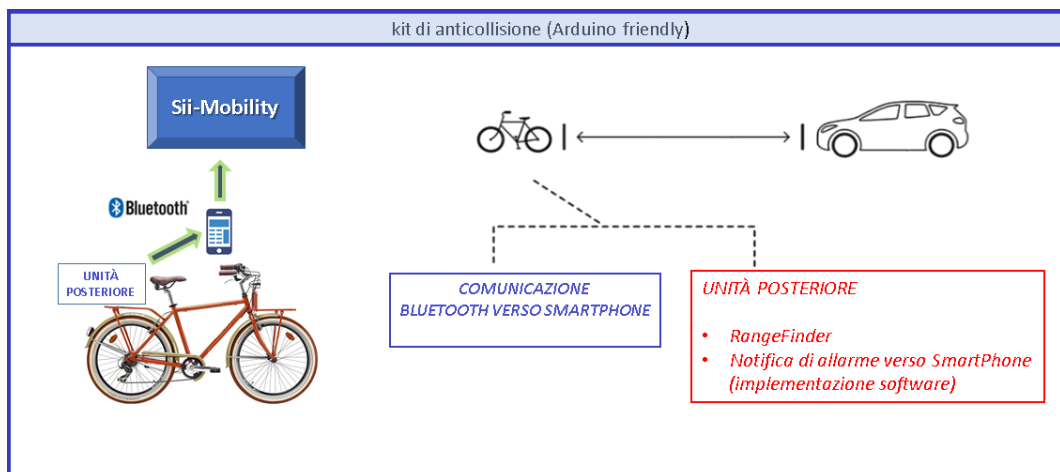


Figura 32: Logica dell'architettura del kit anti collisione

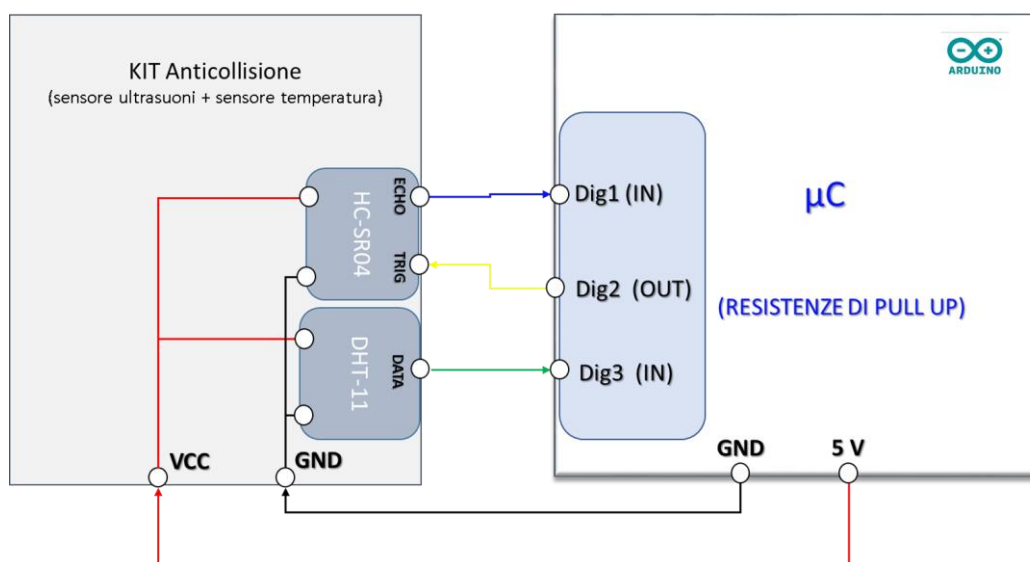


Figura 33: Architettura del sistema (KIT anticollisione)

3.2.4.5.4 Specifica di dettaglio

Sono di seguito riportate alcune caratteristiche aggiuntive del sensore HC SR04, in aggiunta a quelle precedentemente definite, comparandole anche con altri sensori di performance similari: in particolare sono stati analizzati tre sensori ultrasonici i cui coni di apertura del fascio sono illustrati in Figura 34.

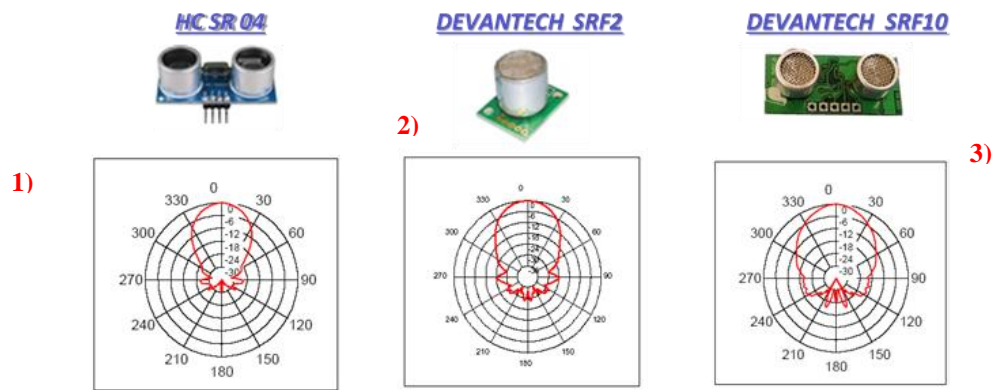


Figura 34: tre sensori acustici scelti come possibile elemento sensibile ai fini del rilevamento della prossimità tra bike e veicolo che segue

La prima soluzione comporta un fascio che offre un'attenuazione di 24 dB per un angolo di 60° rispetto la longitudinale, mentre la terza soluzione è dotata di un fascio molto largo con un'attenuazione di 12 dB a 60°.

La Tabella 7 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.

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 7: 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/giroscopio triassiale

Tabella 8: Caratteristiche principali e peculiari di alcuni dei prodotti Arduino/Genuino presenti sul mercato

Tutte le schede analizzate in Tabella 8 offrono comunicazioni (IIC /SPI /TTL) e livelli di tensione (3.3-5) V compatibili con i sensori scelti. Il sensore **HC SR04** è stato scelto in quanto offre prestazioni ritenute efficaci per una fase iniziale di studio e sperimentazione del sistema anticollisione.

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.

Questa scelta iniziale non preclude la possibilità di acquistare in fase di sperimentazione sensori ultrasonici (*Devantech SRF 10/02*) con caratteristiche diverse sempre nell'ottica di fornire un prodotto il più efficiente possibile a costi contenuti.

3.2.5 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 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.

3.2.5.1 Protocollo di scambio messaggi

In prima istanza, ci si propone di sviluppare il protocollo seguendo lo schema di comunicazione “nativo” Bluetooth Low Energy, descritto in 3.2.1.2.2. 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.

3.2.5.2 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 35](#).

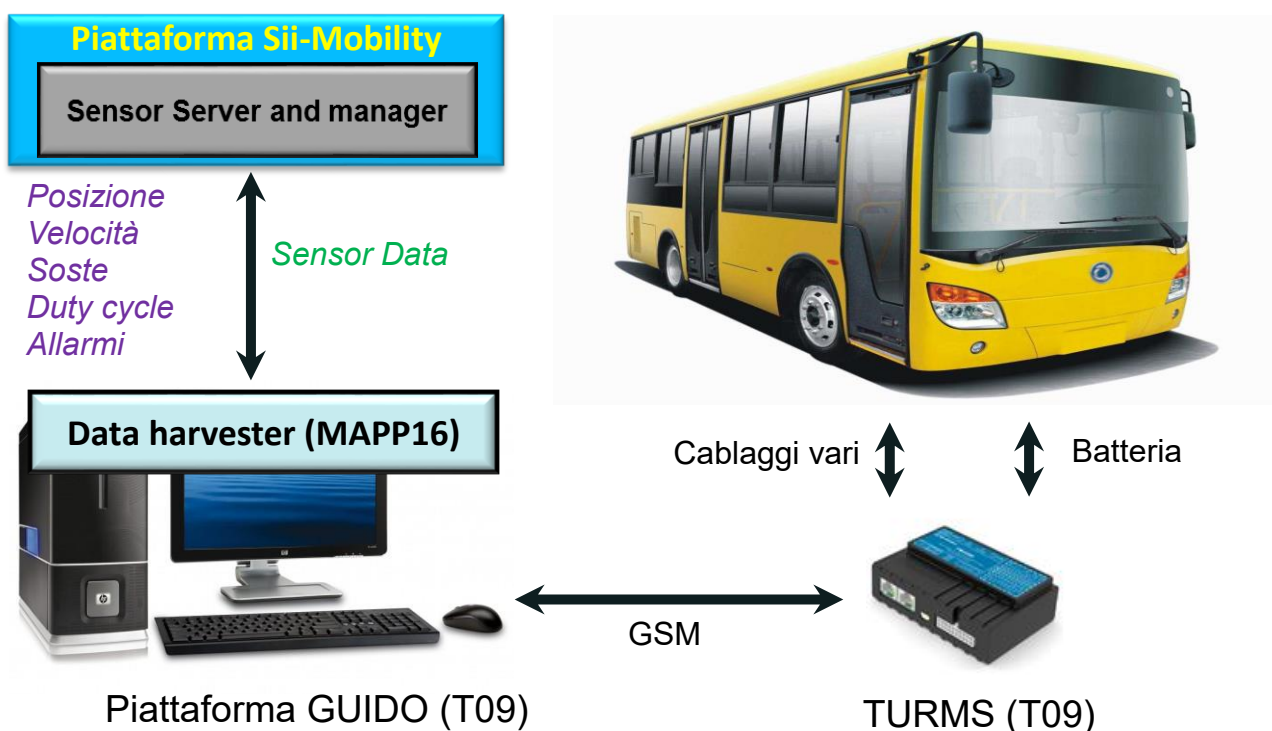


Figura 35 : Architettura generale del Kit carbus base

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:

- **realizzazione** di una versione dedicata del Data Harvester e sua **integrazione** nel software esistente della Piattaforma GUIDO;
- **modifica** del Software esistente della Piattaforma GUIDO per supportare le attività del Data Harvester;
- **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

La [Figura 36](#) 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.

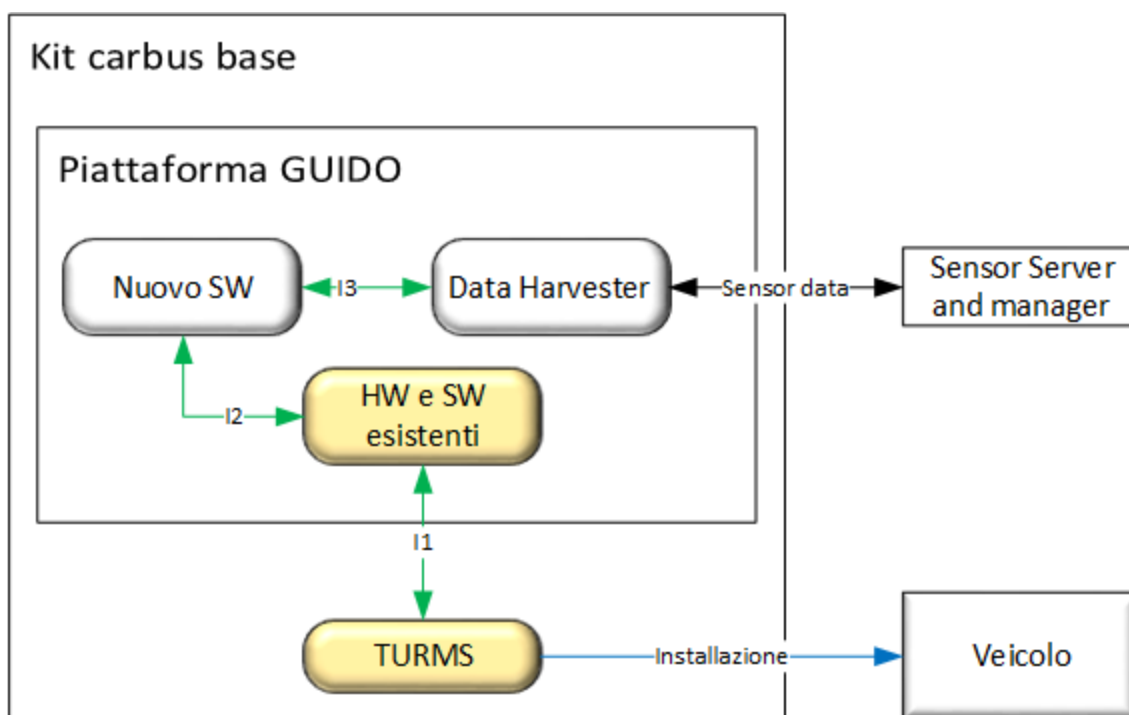


Figura 36 : Kit carbus base - diagramma di contesto

In particolare:

- Interfacce esterne:
 - *Sensor data* è una interfaccia applicativa realizzata con scambi di dati su WEB. Non viene ulteriormente dettagliate in questo documento;
 - *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 0;
- 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;
- **I3** è l'interfaccia fisica e logica tra data harvester e il software esistente sulla Piattaforma GUIDO. L'interfaccia fisica è costituita da servizi del sistema operativo della Piattaforma GUIDO che permettono la comunicazione tra processi. L'interfaccia logica è costituita dai dati scambiati

4.2.1 Interfaccia I3

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.

Per interfacciare il sistema GUIDO con il Data Harvester senza che questo possa interferire con i dati degli altri utenti che hanno installato TURMS ma non fanno parte del progetto Sii-Mobility, si prevede di rendere disponibile una vista del database che includa solamente i TURMS di interesse. Il Data Harvester può interfacciarsi a questo database o con query diretta oppure attraverso sviluppo di apposito back-end.

4.2.2 Nuovo SW

È previsto un nuovo software, da sviluppare a cura di ELFI, che permetta la creazione della vista del database con presenti solo i dispositivi TURMS inclusi in Sii-Mobility, interrogabile dal Data Harvester e, se necessario, un apposito back-end per l'interrogazione.

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

4.2.3 Data Harvester, MAPP16

Il Data Harvester (MAPP16) nelle sue varie istanze e versioni è descritto in altri documenti che al momento non sono determinati. Il riferimento, attualmente "TBD", verrà risolto nella revisione DE3.1b di questo documento a cura di NEGENTIS.

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].



TURMS

CARATTERISTICHE

Alimentazione 7-24VDC

Assorbimento 7 mA* @ 12VDC Modalità LOW POWER

100mA @ 12VDC Modalità Operativa

150mA @ 12VDC GSM in TX

* dipendenti dalle attività GSM e dai comandi inviati

Modalità LOW POWER: Porte 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

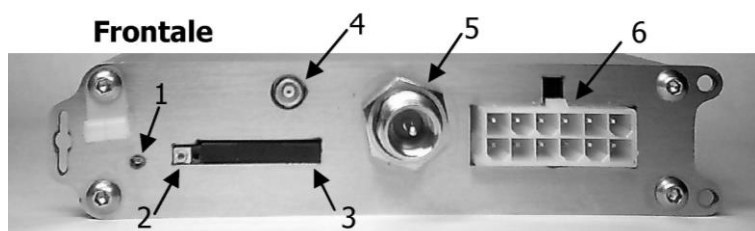
Batteria Ni-Mh 270 mA/h 7,2Vdc Per alimentazione 12V

SISTEMA DI LOCALIZZAZIONE SATELLITARE

MECCANICHE

Involucro in alluminio anodizzato finiture in acciaio inox

Dimensioni 105mmx105mmx26mm



Frontale

1. Led indicatore
2. Pulsante per fuoriuscita cassetto porta SIM
3. Cassetto porta SIM
4. Connettore MCX per antenna GPS
5. Connettore FME per antenna GSM
6. Connettore per cablaggio principale

GPS

-12 canali integrato

-Connettore MCX femmina per RF

GSM

-Connettore FME/maschio per RF

-GSM dual-band EGSM900 e GSM 1800

Compliant GSM phase 2/2+

Output Power

- Classe 4 (2W) per EGSM 900

- Classe 1 (1W) per GSM 1800

-GPRS classe 8 (4+1)

-Voice Call -Data Call -SMS (MT/MO)

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.

Questa sezione sarà completata nella revisione DE3.1b di questo documento a cura di CReAI.

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 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 37*.

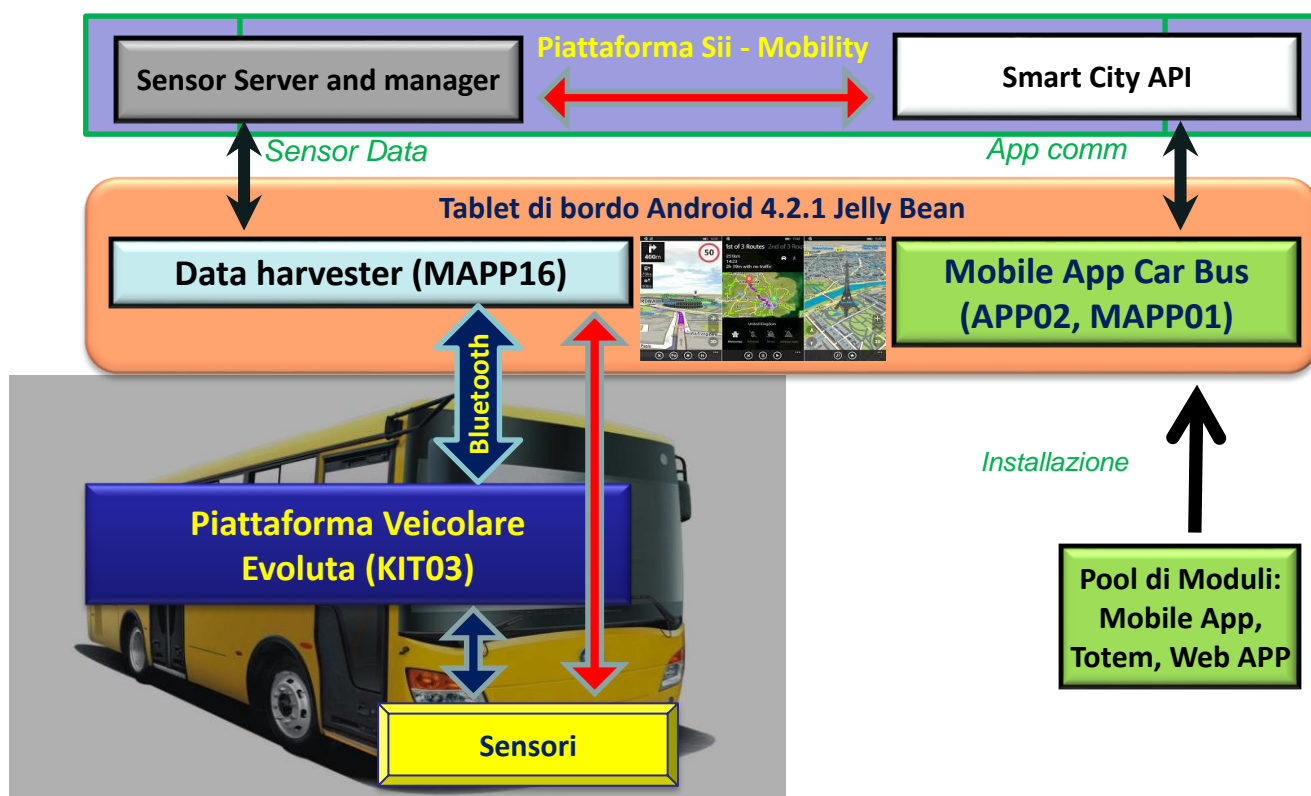


Figura 37 : Architettura generale del Kit carbus evoluto

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,
- se necessario, Data Harvester e Mobile App possono comunicare e sincronizzarsi utilizzando meccanismi standard di Android, ben noti a tutti gli sviluppatori coinvolti,

- 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

La [Figura 38](#) 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.

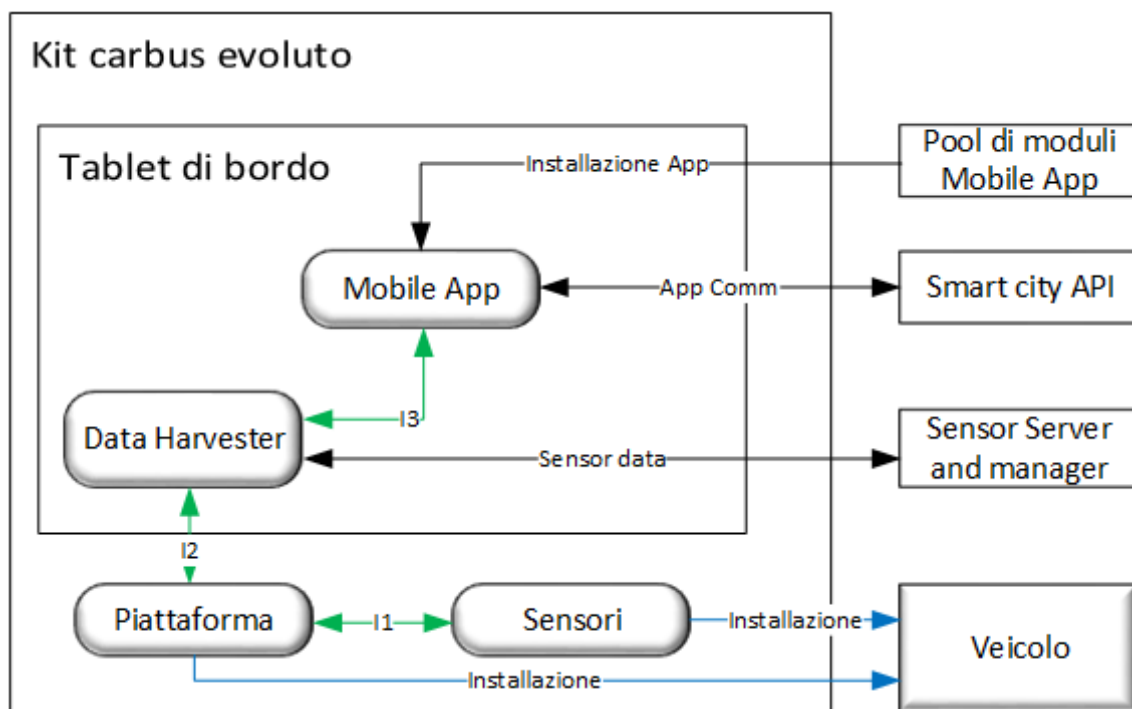


Figura 38 : Kit carbus evoluto - diagramma di contesto

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 5.2.1.1 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;
 - I3 è l'interfaccia fisica e logica tra Mobile App e Data Harvester.

5.2.1 Piattaforma

La [Figura 39](#) 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.

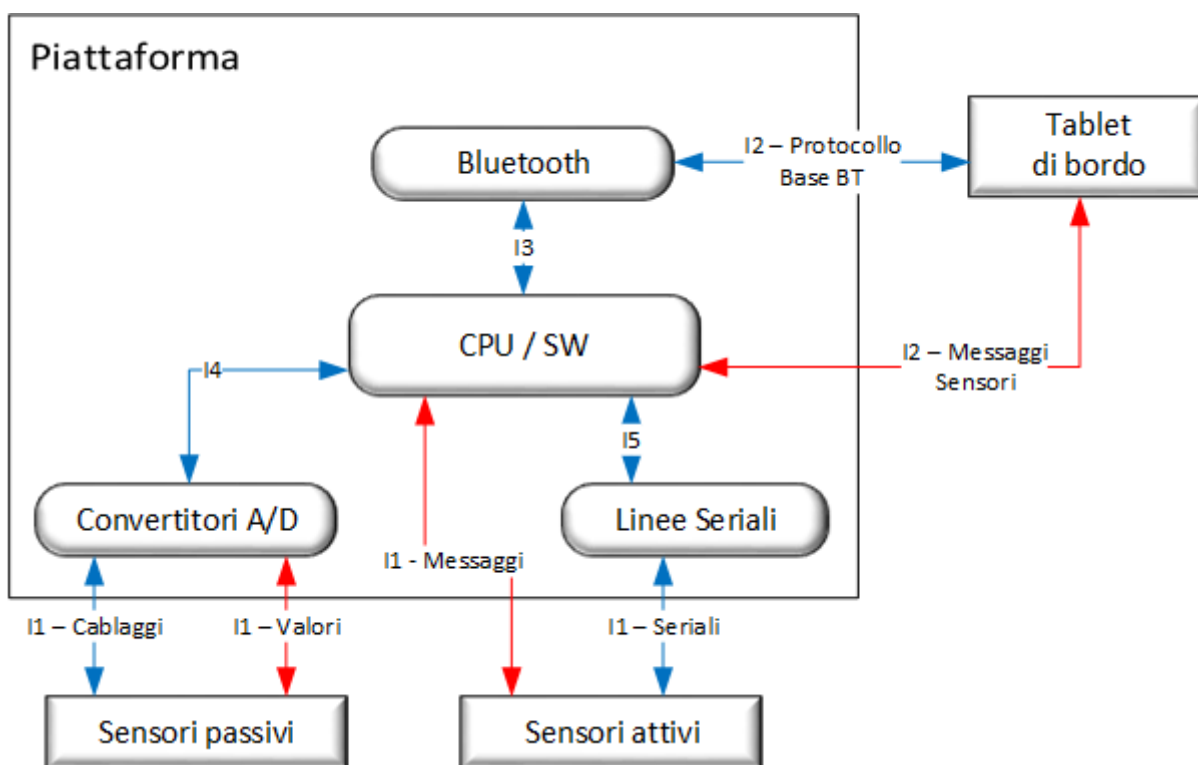


Figura 39 : Piattaforma – diagramma di contesto

La piattaforma include gli elementi seguenti, tutti descritti in dettaglio alla sezione 5.2.1.2:

- 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;
- un dispositivo Bluetooth per le comunicazioni con il Tablet;
- una scheda CPU (motherboard).

I sensori e apparati interfacciati sono:

- uno “Smart Node” che raccoglie i messaggi dalle telecamere presenti a bordo:
 - telecamera per conteggio passeggeri (5.2.3.2),
 - telecamera per monitoraggio traffico e ambientale (5.2.3.3),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 5.2.3.1
- sensori ambientali, descritti in dettaglio alla sezione 5.2.3.4

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 (NEGENTIS) 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 43*). 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*

5.2.1.1 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.

Questa sezione sarà completata nella revisione DE3.1b di questo documento a cura di CReAI.

5.2.1.2 Hardware

5.2.1.2.1 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 3.2.1.2.1 e 3.2.1.2.2.

5.2.1.2.2 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 (5.2.3.2, 5.2.3.3), sensori ambientali (5.2.3.4), un modulo CanBus sniffer (5.2.3.1).

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 40 e la Tabella 9 mostrano uno schema generale del kit e il dettaglio del collegamento delle interfacce fisiche alle porte di Arduino 101.

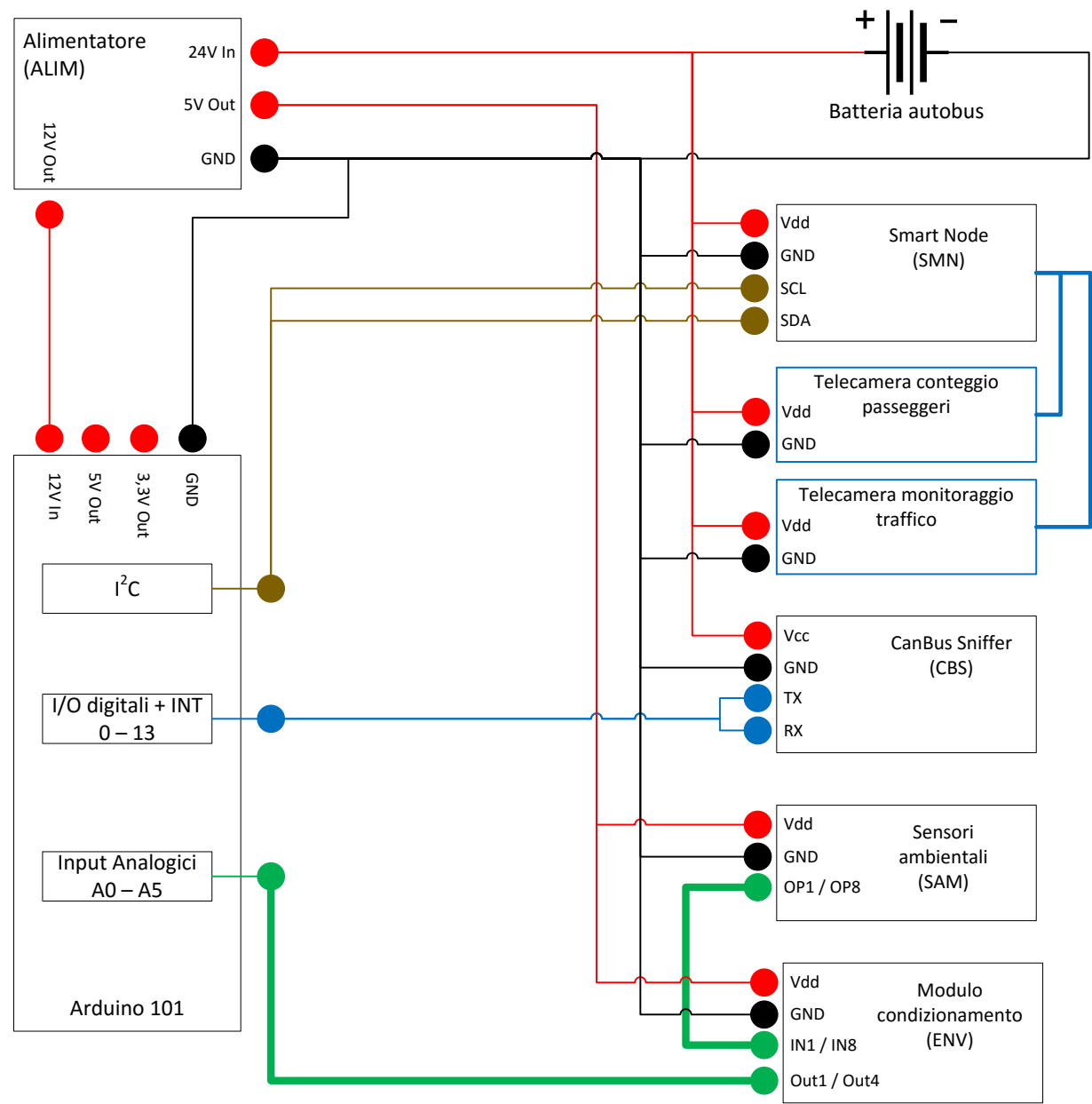


Figura 40 : Schema generale del kit carbus evoluto

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 9: Kit carbus evoluto - connessioni fisiche tra motherboard e sensori

*: porte che supportano PWM

°: porte che supportano interrupt sui cambiamenti di stato

5.2.1.3 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.

5.2.1.3.1 Casi d'uso

In Figura 41 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;
 - **Attori:** Time, Piattaforma
 - **Pre-condizione:** UC1 completato;
 - **Sequenza degli eventi:** secondo le specifiche del Protocollo base BT (Sezione 5.2.4):
 1. Il CU inizia quando scade il Timer periodico di data sending;
 2. La Piattaforma invia al Tablet un messaggio contenente i dati di misura dei sensori acquisiti nel caso d'uso UC2;
 3. La Piattaforma imposta il Timer periodico di data sending;
 - **Post-condizione:** la Piattaforma ha inviato il messaggio al Tablet ed il Timer periodico è attivo.
- **Clock synchronization:** coinvolge l'interfaccia I2. Sincronizzazione dei clock tra Piattaforma e Tablet
 - **ID:** UC4;
 - **Attori:** Piattaforma, Time, Tablet
 - **Pre-condizione:** UC1 completato;
 - **Sequenza degli eventi:** secondo le specifiche del Protocollo base BT (Sezione 5.2.4):
 1. Il CU inizia quando scade il Timer periodico di sincronizzazione dei clock;
 2. La Piattaforma invia al Tablet un messaggio di clock synchronization;
 3. Il Tablet risponde con il clock;
 4. La Piattaforma imposta il Timer di sincronizzazione;
 - **Post-condizione:** i clock sono sincronizzati ed il Timer periodico è attivo.
- **Vehicular deassociation:** deassociazione tra Piattaforma e Tablet.
 - **ID:** UC5;
 - **Attori:** Piattaforma, Tablet
 - **Pre-condizione:** UC1 completato;
 - **Sequenza degli eventi:** secondo le specifiche del Protocollo base BT (Sezione 5.2.4):
 1. Il CU inizia quando la Piattaforma riceve dal Tablet un messaggio di deassociazione;
 2. La Piattaforma risponde al Tablet;
 - **Post-condizione:** Piattaforma e Tablet sono deassociati ed il UC1 è attivo.

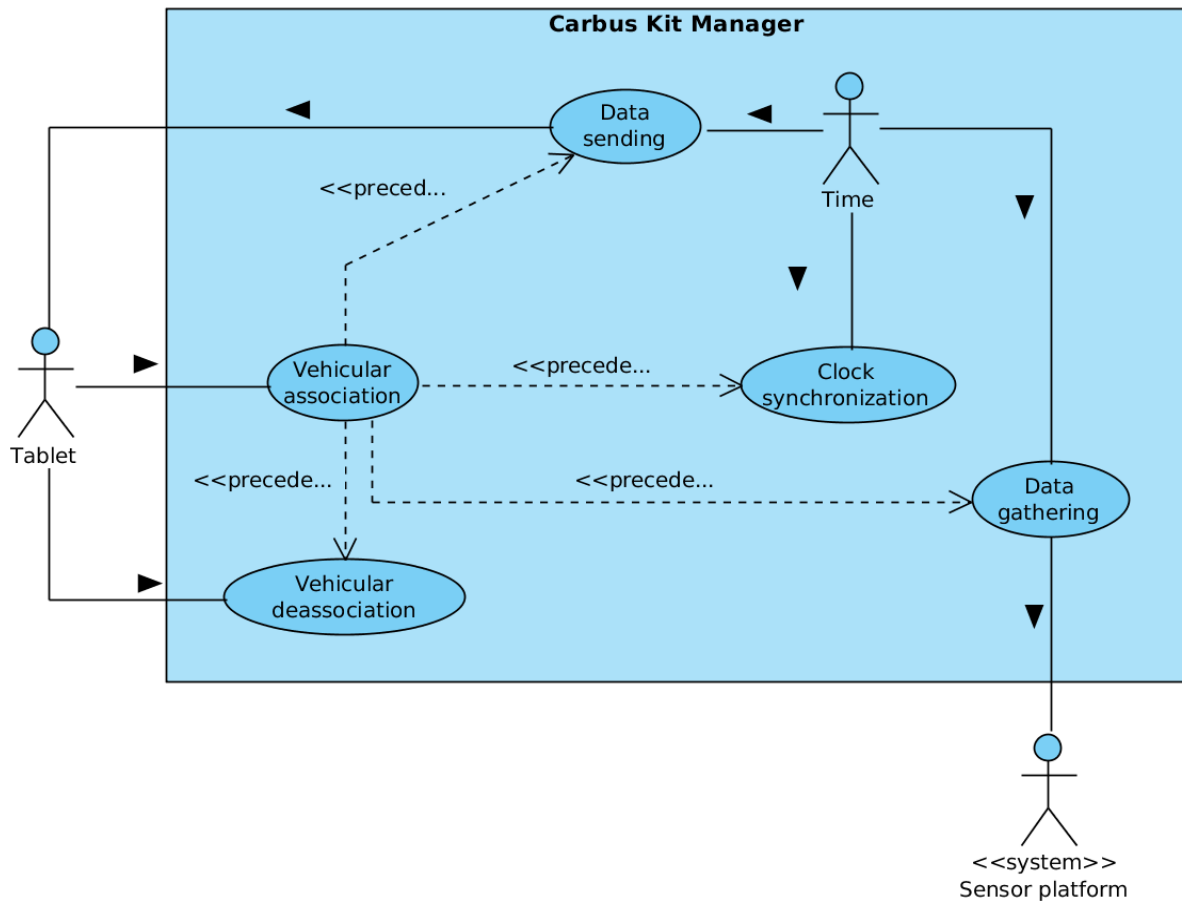


Figura 41 : Carbus Kit Manager Bike – casi d’uso

5.2.1.3.2 Macchina a stati

In Figura 42 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`.

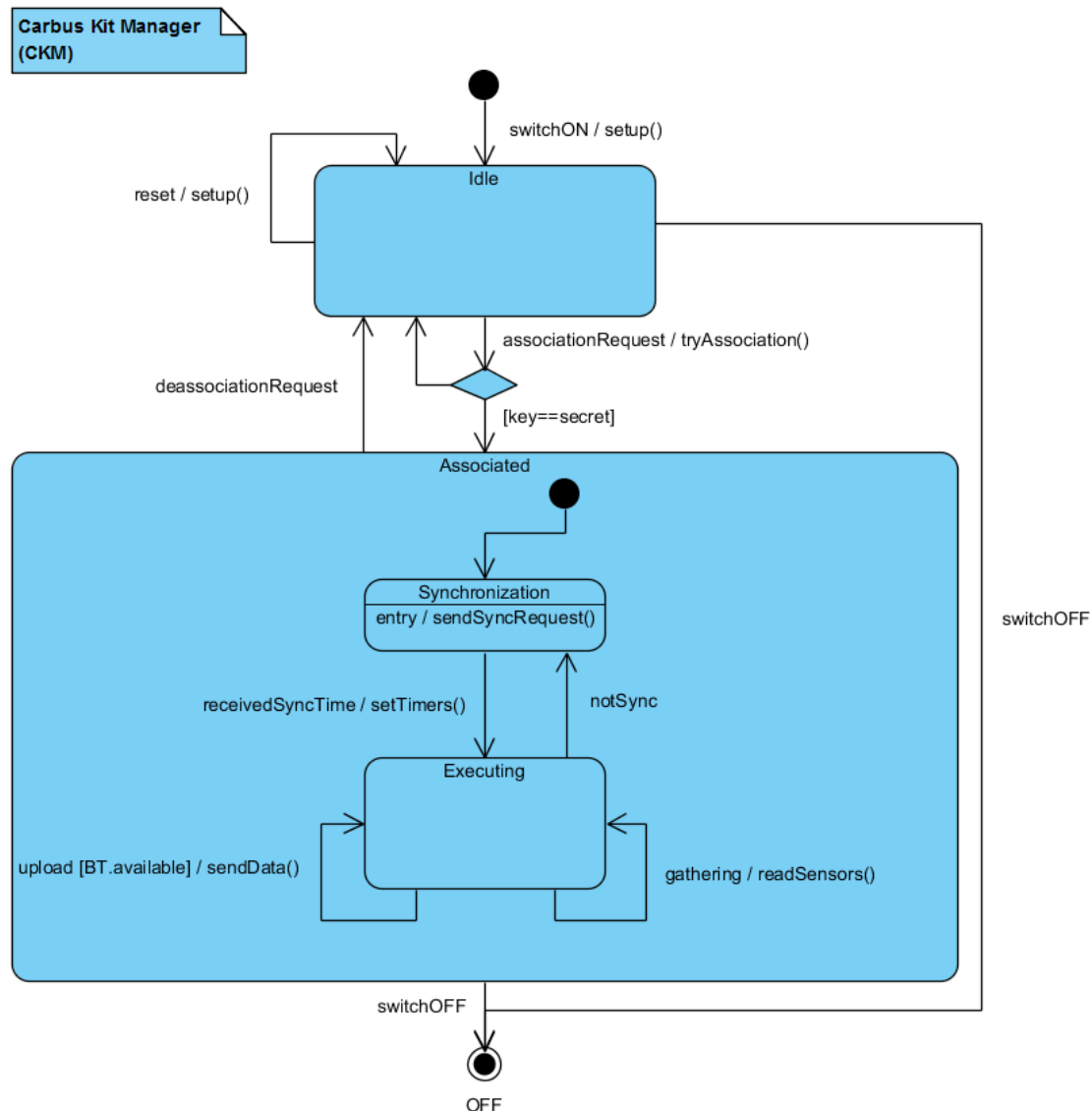


Figura 42: Carbus Kit Manager Bike – FSM

5.2.1.3.3 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 5.2.1.3.2 per realizzare i casi d’uso descritti in 5.2.1.3.1. 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 5.2.4 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.

5.2.2 Tablet di bordo

La [Figura 43](#) mostra il diagramma di contesto dello Smartphone Utente

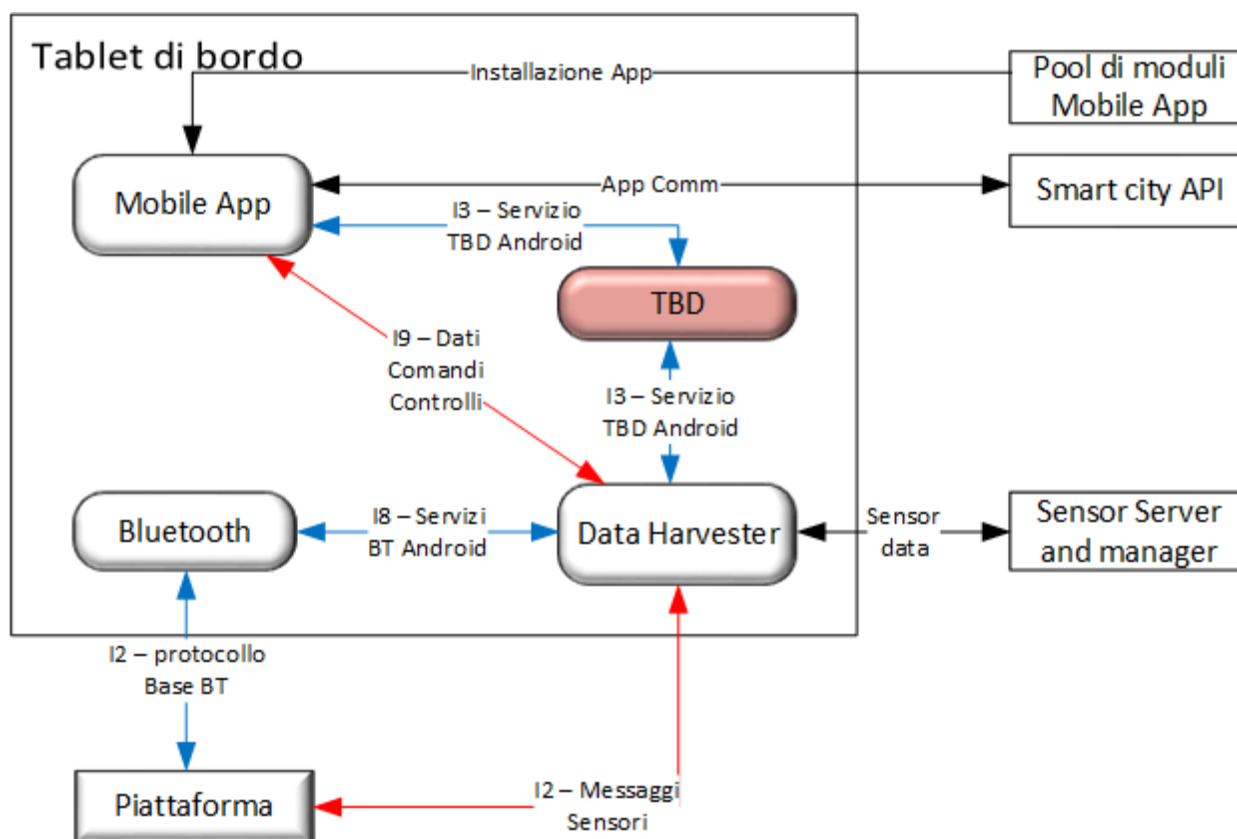


Figura 43 : Smartphone – diagramma di contesto

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

5.2.2.1 Data Harvester, MAPP16

Il Data Harvester (MAPP16) nelle sue varie istanze e versioni è descritto in altri documenti che al momento non sono determinati. Il riferimento, attualmente “TBD”, verrà risolto nella revisione DE3.1b di questo documento a cura di NEGENTIS.

5.2.2.2 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. Il riferimento, attualmente “TBD”, verrà risolto nella revisione DE3.1b di questo documento a cura di DISIT.

APP02 incorpora MAPP01 che esegue il monitoraggio di situazioni critiche analizzando i dati accelerometrici del Tablet di bordo. MAPP01 è realizzata a cura di MIZAR e descritta nella sezione seguente.

5.2.2.3 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.

5.2.3 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.

5.2.3.1 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.

5.2.3.1.1 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 44).

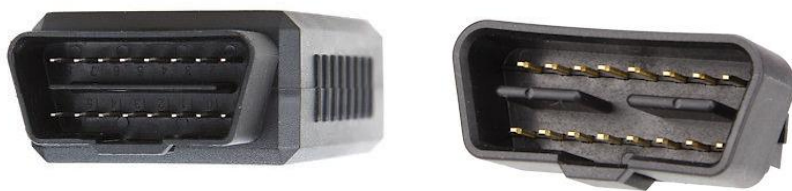


Figura 44 : Connettori OBD-II – tipo “a” (sinistra) e tipo “b” (destra)

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 45).

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 45 : 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.

5.2.3.1.2 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.

5.2.3.1.3 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 descritta nella revisione DE3.1b di questo documento a cura di CReAI.

5.2.3.2 Telecamera Intelligente per conteggio passeggeri, SN05

Di seguito i riferimenti nel deliverable dedicato ai sensori innovativi ([SiiM 4]).

5.2.3.2.1 Interfacce

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

- l'interfaccia fisica è descritta alla sezione 2.2.1.2
- l'interfaccia logica è descritta alla sezione 2.2.1.3

5.2.3.2.2 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 descritte nella revisione DE3.1b di questo documento a cura di PROJECT.

5.2.3.3 Telecamera Intelligente per monitoraggio traffico e ambientale, SN05

Di seguito i riferimenti nel deliverable dedicato ai sensori innovativi ([SiiM 4]).

5.2.3.3.1 Interfacce

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

- l'interfaccia fisica è descritta alla sezione 2.2.4.2
- l'interfaccia logica è descritta alla sezione 2.2.4.3

5.2.3.3.2 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 descritte nella revisione DE3.1b di questo documento a cura di PROJECT.

5.2.3.4 Sensori ambientali

Si tratta di sensori dedicati 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 congiunzioni,

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

5.2.3.4.1 Interfacce

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

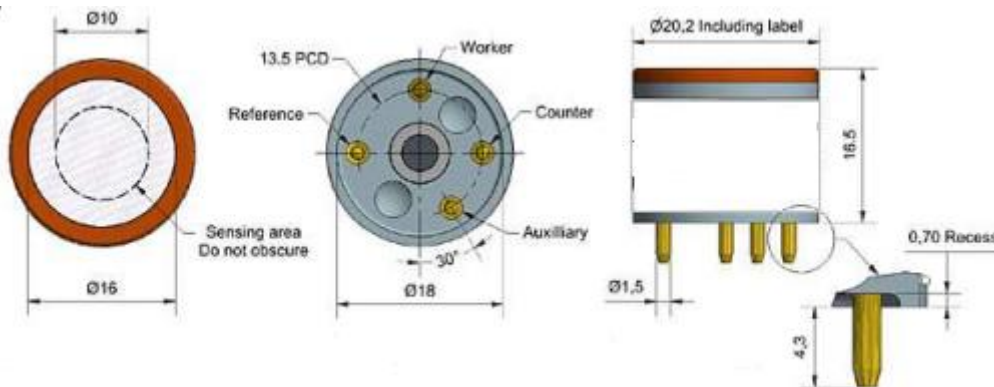
- l'interfaccia fisica è descritta alla sezione 3.1.1.2 e 3.2.3.2.
- l'interfaccia logica è descritta alla sezione 3.1.1.3 e 3.2.3.3

5.2.3.4.2 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.

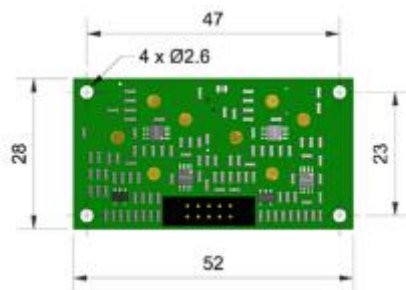
Nell'un caso e nell'altro dovranno essere previsti degli scassi che consentano ai sensori di poter entrare in contatto con l'aria in modo da poter rilevare la presenza degli eventuali inquinanti.

Si riportano di seguito le dimensioni fisiche dei sensori

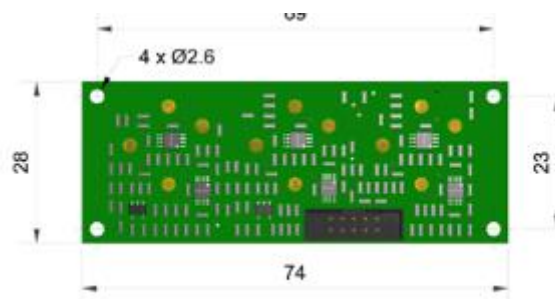


Sensore di tipologia A4: Dimensioni Meccaniche

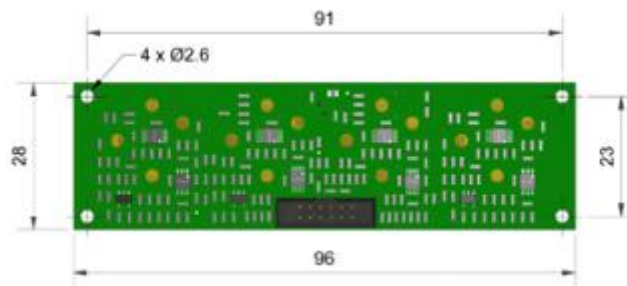
E le dimensioni delle AFE



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 5.2.1.1.

5.2.4 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 39](#) e [Figura 43](#).

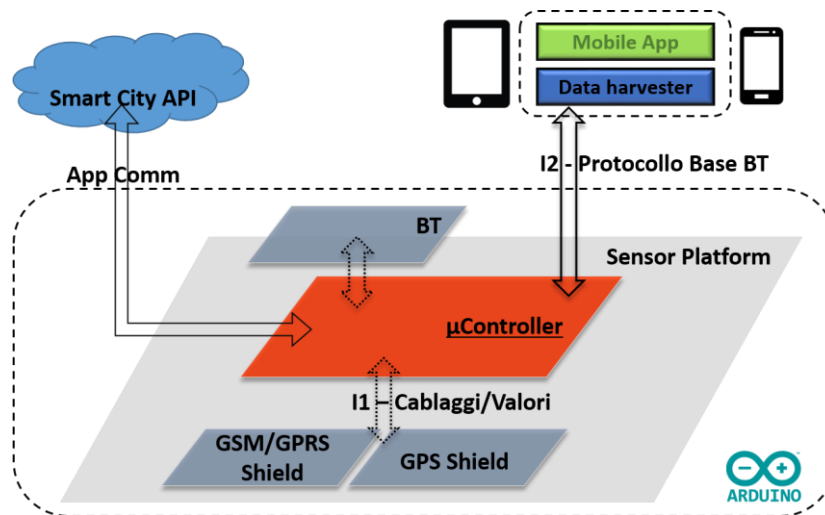
La specifica deve essere condivisa tra chi sviluppa il software della Piattaforma e del Data Harvester, in modo da utilizzare in modo consistente le funzionalità Bluetooth presenti sulla Piattaforma e sul Tablet per realizzare un meccanismo di trasporto wireless di messaggi.

In prima istanza, ci si propone di sviluppare il protocollo seguendo lo schema di comunicazione “nativo” Bluetooth Low Energy, descritto in 3.2.1.2.2. 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 (LART)

La funzione Autonomous Anti-Theft Feature (AATF), di cui alla Sezione 3.2.2 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 3.2.2 - scenario b.1.i e Sezione 3.2.4.2. ovvero la soluzione AATF con GPS e segnale sonoro di allarme, Sezione 3.2.4.3. 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.