



Sii-Mobility

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

Trasporti e Mobilità Terrestre, SCN_00112

Deliverable ID: DE3.11b

**Titolo: Attuatori integrati - specifica di dettaglio dello
sviluppo**

Data corrente	M17, Maggio 2017
Versione (solo il responsabile può cambiare versione)	v1-12
Stato (draft, final)	Final
Livello di accesso (solo consorzio, pubblico)	Pubblico quando completo
WP	OR3
Natura (report, report e software, report e HW..)	report
Data di consegna attesa	M17, Maggio 2017
Data di consegna effettiva	M17, Maggio 2017
Referente primario, coordinatore del documento	Alessandro Paganone, ECM, alessandro.paganone@ecmre.com
Contributor	Leonardo Fabbri, ELFI, leonardo.fabbri@elfisrl.net Andrea Rindi, UNIFI-DIEF/Mdmlab, andrea.rindi@unifi.it Argeo Bartolomei, ARGOS, abartolomei@argosengineering.it

	Paolo Nesi, paolo.nesi@unifi.it	UNIFI,
Coordinatore responsabile del progetto	Paolo Nesi, paolo.nesi@unifi.it	UNIFI,

Sommario

1	Executive Summary	6
2	Contesto, terminologia, riferimenti e notazioni	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
3	Segnale Dinamico	10
3.1	Struttura.....	10
3.2	Scenario di utilizzo.....	11
3.3	Architettura generale	11
3.4	Specifica di dettaglio	13
3.4.1	Diagramma di contesto generale.....	13
3.4.2	Piattaforma, EHW01	14
3.4.3	Tablet di servizio.....	20
3.4.4	Sensori / Attuatori	20
3.4.5	Protocollo base Bluetooth	22
4	Direzionatore.....	23
4.1	Struttura.....	23
4.2	Scenario di utilizzo.....	23
4.3	Architettura generale	24
4.4	Specifica di dettaglio	25
4.4.1	Diagramma di contesto generale.....	25
4.4.2	Piattaforma, EHW03	25
4.4.3	Tablet di servizio.....	26
4.4.4	Sensori / attuatori	26
4.4.5	Protocollo base Bluetooth	26
5	Rastrelliera Intelligente	27
5.1	Struttura.....	27
5.2	Scenario di utilizzo.....	27
5.3	Architettura generale	29
5.4	Specifica di dettaglio	31

5.4.1	Diagramma di contesto generale.....	31
5.4.2	Piattaforma, EHW05	32
5.4.3	Sensori / attuatori	40
Figura 1	: Attuatori Integrati nel contesto di Sii-Mobility	8
Figura 2	: notazione grafica generale	9
Figura 3	: notazione grafica di dettaglio	9
Figura 4	: Struttura di base del Segnale Dinamico.....	10
Figura 5	: Scenario di utilizzo del Segnale Dinamico.....	11
Figura 6	: Architettura generale del Segnale Dinamico	12
Figura 7	: Segnale Dinamico - diagramma di contesto generale	13
Figura 8	: Segnale Dinamico – diagramma di contesto della Piattaforma.....	14
Figura 9	: Schema generale del Segnale Dinamico.....	18
Figura 10	: Segnale Dinamico - architettura di base del software	19
Figura 11	: Segnale Dinamico – diagramma di contesto del Tablet di servizio	20
Figura 12	: Scenario di utilizzo del Direzionatore	23
Figura 13	: Architettura generale del Direzionatore	24
Figura 14	: Descrizione dell'architettura di controllo e del singolo stallo della rastrelliera.....	27
Figura 15	: Scenario 1- Utente viene aggiornato dello stato della rastrelliera tramite interrogazione del SSM.....	28
Figura 16	: Scenario 2- Trasmissione della richiesta di una bike e rilascio di questa a seguito di comando di sblocco da piattaforma	28
Figura 17	: Scenario 3- Al momento del rilascio della bike il sistema intelligente a bordo rastrelliera notifica la piattaforma la quale provvede all’aggiornamento dello stato comunicandolo al SSM	29
Figura 18	: Architettura generale della Rastrelliera Intelligente.....	30
Figura 19	: Attraverso la lettura periodica dei tag è possibile rilevare la mancata presenza di una bike che, se non preceduta da una autorizzazione da parte del SSM, innesca il meccanismo di allarme a bordo della piattaforma	31
Figura 20	: Rastrelliera Intelligente - diagramma di contesto generale	31
Figura 21	: Rastrelliera Intelligente - diagramma di contesto della Piattaforma	32
Figura 22	: Collegamenti tra la motherboard, sensori RFID (vedi 5.4.3.1) e attuatori (vedi 5.4.3.2.1.3)	34
Figura 23	: Modulo Shield GSM/GPRS e connessioni con Arduino UNO	35
Figura 24	: Schema generale della Rastrelliera Intelligente-definizioni delle connessioni per la parte di raccolta dati proveniente dai sistemi RFID di ogni stallo.....	37

Figura 25 : Schema generale della Rastrelliera Intelligente-definizioni delle connessioni di attuazione dei solenoidi per aggancio/sgancio della bicicletta allo stallo.....	37
Figura 26 : Realizzazione di un Web Service secondo architettura in stile REST	38
Figura 27 : Sistema locale di identificazione RFID installato sullo stallo della rastrelliera.....	41
Figura 28 : Sistema locale di identificazione RFID - PN 532 RFID/ NFC (Arduino like)	41
Figura 29 : definizione della fase di START/STOP, ACKNOWLEDGE sul bus I2C	43
Figura 30 : comunicazione sul bus I2C (indirizzo slave + data)	43
Figura 31 : Architettura generale di principio del sistema elettronico, caso operativo di rastrelliera con singolo stallo	44
Figura 32 : CHIP P82B715 per estendere il bus I2C	44
Figura 33 : perno azionamento con intervento esterno (Sx) e elettromagnete dotato di bobina che permette lo sblocco mediante applicazione di una tensione di 12 V (Dx).....	45
Figura 34 : cursore in posizione di blocco mediante compressione a molla (Sx) e cursore in posizione di sblocco mediante applicazione di eccitazione della bobina tramite una tensione di 12 V	45
Figura 35 : schema di principio di controllo tramite pin digitale di Arduino del singolo solenoide responsabile del gancio/sgancio della bicicletta allo stallo.....	46
Figura 36 : finecorsa ottico SHARP GP1A57HRJ00F	46
Figura 37 : breakout board per l'implementazione del sensore di finecorsa ottico SHARP GP1A57HRJ00F	47
Figura 38 : Datasheet del sensore SHARP GP1A57HRJ00F-distanza tra emettitore e ricevitore	47
Figura 39: Schema generale della Rastrelliera Intelligente-definizioni delle connessioni di lettura del segnale dei finecorsa	48
Figura 40 : relè dedito all'azionamento del perno usato per l'aggancio/sgancio della bicicletta alla rastrelliera.....	48
Figura 41: solenoide tubolare a trazione Saia-Burgess (modello 195222-23).....	49

1 Executive Summary

Il documento contiene il progetto dettagliato degli Attuatori Integrati previsti nel contesto del progetto:

- Segnale Dinamico - per le porte di accesso ai parcheggi e alle ZTL;
- Direzionatore - per la modifica dinamica del senso di percorrenza di una strada o corsia;
- Rastrelliera Intelligente - per realizzare la funzione **AATF** (autonomous anti-theft feature).

Tutti gli attuatori integrati sono realizzati interamente nel contesto del progetto. Per ciascuno di essi 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]/[SiiM 11]).

Questo documento (DE3.11b) ha lo scopo di descrivere quanto si intende sviluppare come prototipo al termine delle fasi di progetto. Altri documenti sono previsti per monitorare lo sviluppo dei prototipi ([SiiM 9]), per fornire la manualistica di configurazione e installazione ([SiiM 10]) e i risultati della sperimentazione ([SiiM 12]). Lievi variazioni in corso d'opera saranno descritte direttamente nei documenti ora menzionati, in caso di variazioni di maggior rilievo sarà possibile una ri-emissione di questo documento.

2 Contesto, terminologia, riferimenti e notazioni

2.1 Acronimi, sigle, terminologia

API	Application Program Interface
CEN	European Committee for Standardization
GPRS	General packet radio service
GPS	Global positioning System
GSM	Global System for Mobile
HDMI	High Definition Multimedia Interface
HTML	Hyper Text Markup Language
ICT	Information and Communication Technologies
LCD	liquid-crystal display
OD	Open Data
RFID	Radio Frequency IDentification o Identificazione a radio frequenza
SII	sistema di interoperabilità integrato

SSAMM	Agenzia per la Mobilità Metropolitana strumenti di supporto, TOSCANA
TPL	gestore trasporto pubblico locale
UMTS	Universal Mobile Telecommunications System
V2I	Vehicle-to-Infrastructure
V2V	vehicle-to-vehicle
WSDL	Web Services Description Language
WSN	Wireless Sensor Networks
ZTL	Zona a Traffico Limitato

2.2 Documenti di riferimento

2.2.1 Standards e normative applicabili

[Std 1]	TBD

2.2.2 Documenti Sii-Mobility

[SiiM 1]	DE1.1a v3-0 - Analisi dei requisiti e casi d'uso
[SiiM 2]	DE1.2a v4-5 - Specifica di Integrazione e Casi di Test
[SiiM 3]	DE8.5 v0-6 - Manuale di qualità del progetto
[SiiM 4]	DE3.6a v8 – Sensori : specifica di dettaglio dello sviluppo
[SiiM 5]	DE3.11a v0-11 – Attuatori Integrati: specifica di dettaglio dello sviluppo
[SiiM 6]	Sii-Mobility-meeting-2016-11-28-v7-0 (PPT), verbale della riunione
[SiiM 7]	2016.09.16 Verbale riunione skype OR3
[SiiM 8]	DE3.1b – Kit veicolari : specifica di dettaglio dello sviluppo
[SiiM 9]	DE3.12 – Attuatori integrati: prototipi
[SiiM 10]	DE3.14 – Attuatori Integrati : manuale di installazione e programmazione
[SiiM 11]	DE3.6b – Sensori : specifica di dettaglio dello sviluppo
[SiiM 12]	DE3.13 - Attuatori integrati: rapporto di sperimentazione
[SiiM 13]	DE3.11b - Allegato1 - Segnale Dinamico - Direzioneatore
[SiiM 14]	DE3.11b - Allegato2 - Rastrelliera Intelligente

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)

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 3.3 - Studio, definizione e sviluppo di Attuatori integrati per controllo accessi, direzione e velocità*.

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

La *Figura 1* mostra il ruolo degli Attuatori Integrati nell'architettura generale di Sii-Mobility. Complessivamente, gli attuatori integrati hanno lo scopo di:

- interagire con *Sensori e Attuatori* per acquisirne i dati e comandarli / monitorarli,
- ospitare Applicazioni SW (*Processi*) fornendo loro i servizi per svolgere le loro funzioni, includendo anche le azioni descritte al punto precedente.

I Processi interagiscono con i sistemi informativi dei Gestori di parcheggi e/o ZTL, oppure direttamente con la piattaforma Sii-Mobility.

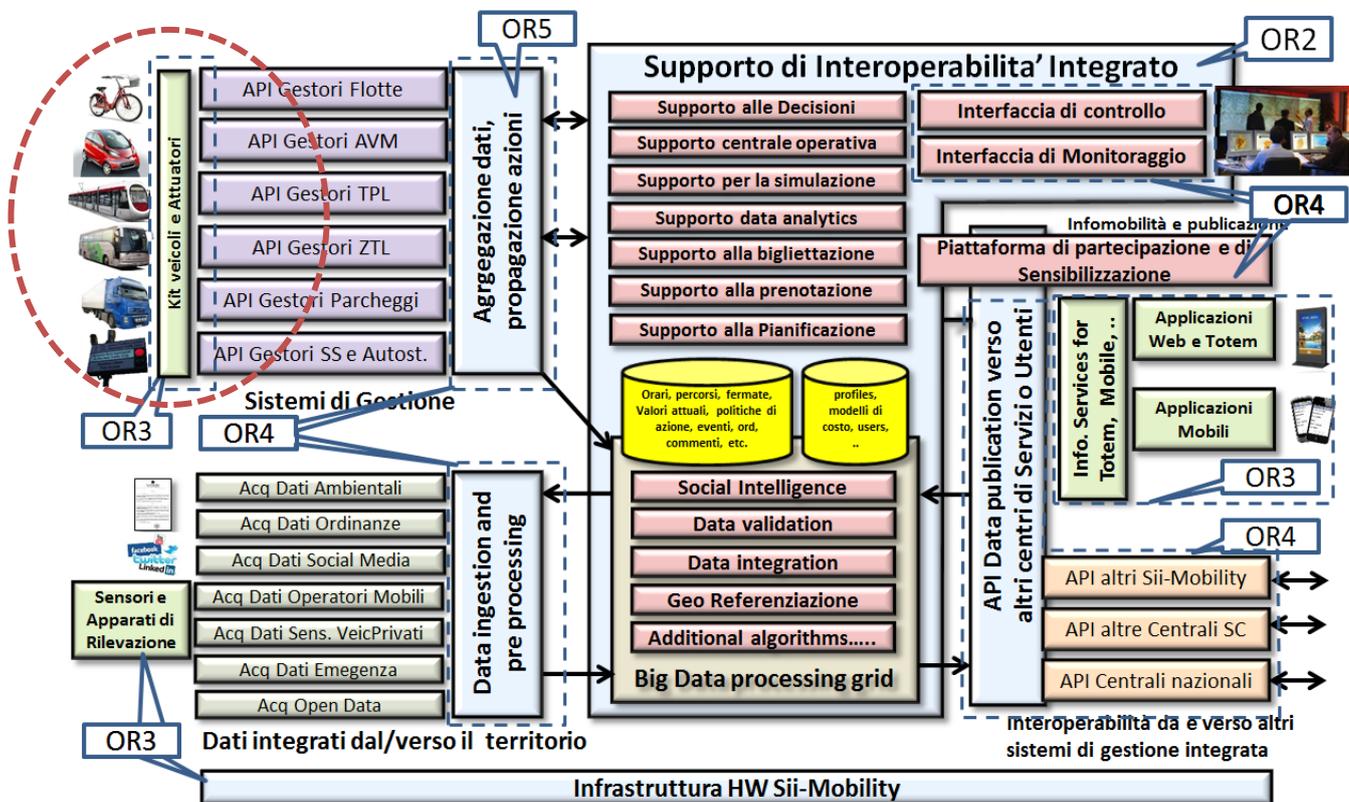


Figura 1 : Attuatori Integrati nel contesto di Sii-Mobility

2.4 Notazioni

2.4.1 Schemi di architettura

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

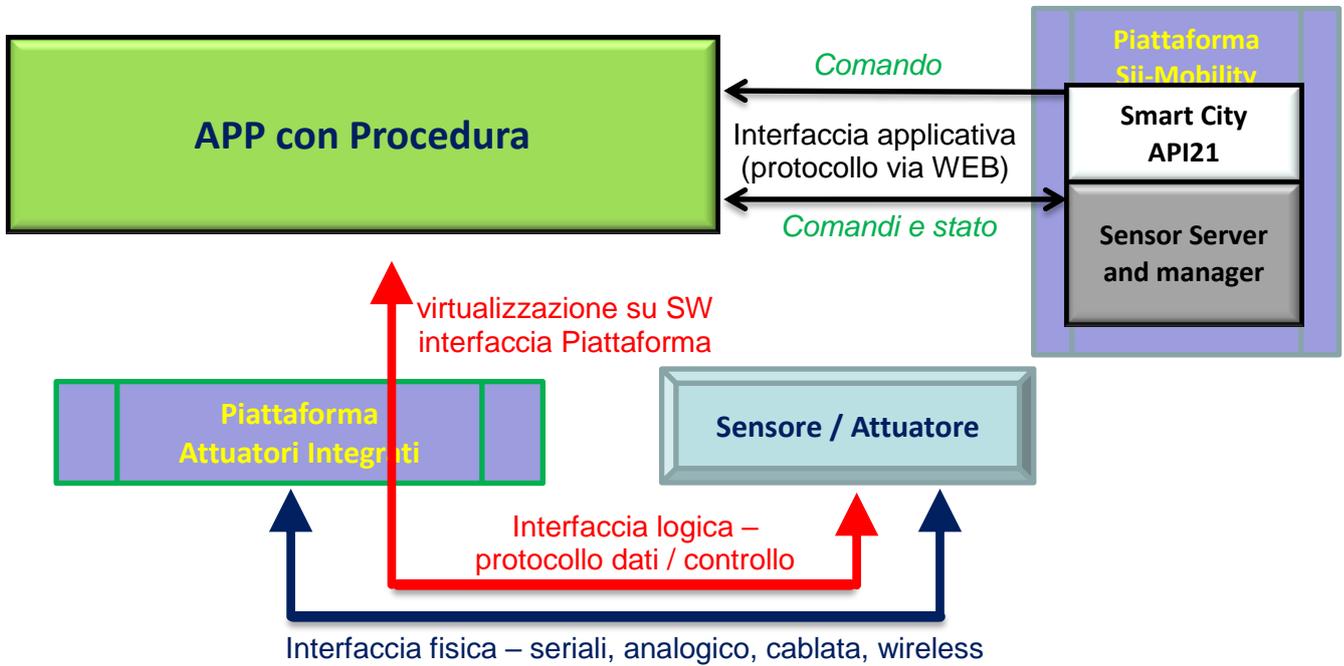


Figura 2 : notazione grafica generale

2.4.2 Schemi di dettaglio

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

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

Figura 3 : notazione grafica di dettaglio

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

3 Segnale Dinamico

3.1 Struttura

Il Segnale Dinamico 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* (e inoltre ne può contenere altri).

Altro componente fondamentale è una APP che realizza una *Procedura* in grado da un lato di comunicare con la centrale Sii-Mobility per scambiare dati, comandi e notifiche e dall'altro di interfacciarsi con la Piattaforma e quindi con Sensori e Attuatori.

Per lo sviluppo e il supporto della Procedura si utilizza un *Tablet* di servizio, in modo da fornire un host espressamente pensato per applicazioni WEB based. Il Tablet è componente sostanziale del Segnale Dinamico.

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

La *Figura 4* mostra la strutturazione descritta.

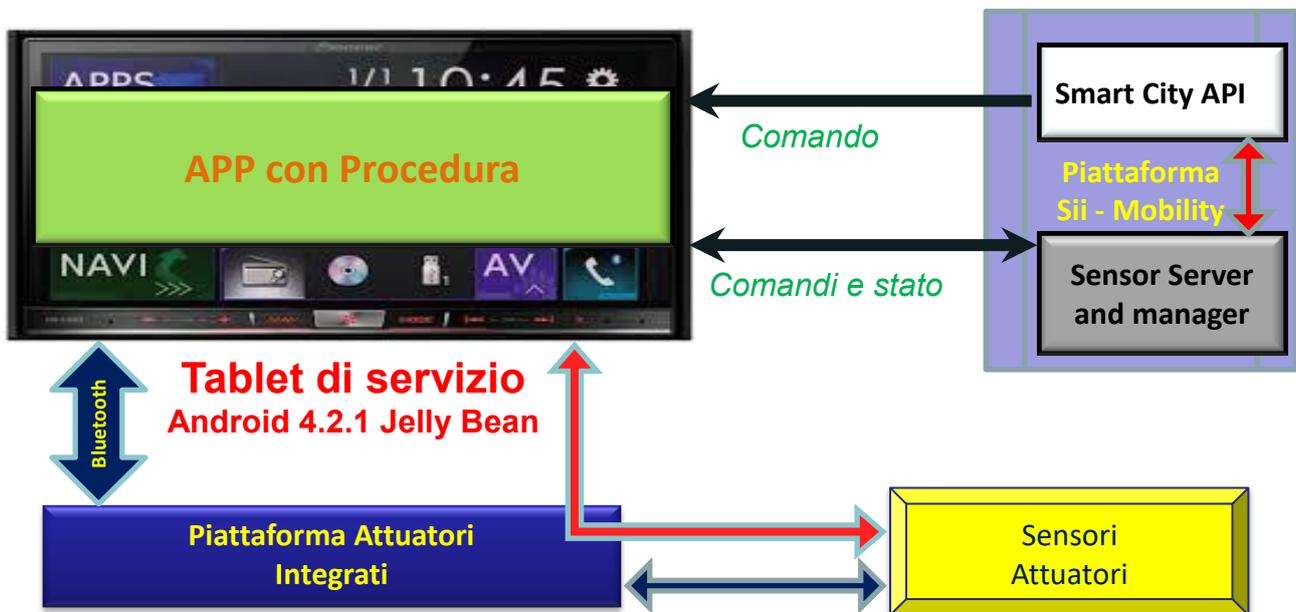


Figura 4 : Struttura di base del Segnale Dinamico

I vantaggi offerti da questa organizzazione sono molteplici e rilevanti:

- chi sviluppa la Procedura opera in un ambiente “familiare”, costituito da un apparato commerciale, con caratteristiche ben standardizzate. Utilizza una installazione originale del sistema operativo Android, mentre un “porting” di Android sulla Piattaforma esporrebbe a rischi di comportamenti anomali,
- il canale di comunicazione Bluetooth può essere usato efficacemente sia per rendere disponibili alla Procedura i dati dei sensori contenuti nella Piattaforma stessa, sia per virtualizzare l’interazione con i sensori esterni alla piattaforma. Lo stesso vale per i comandi da inviare agli attuatori,

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

3.2 Scenario di utilizzo

Il tipico scenario di utilizzo del Segnale Dinamico è mostrato nella *Figura 5*:

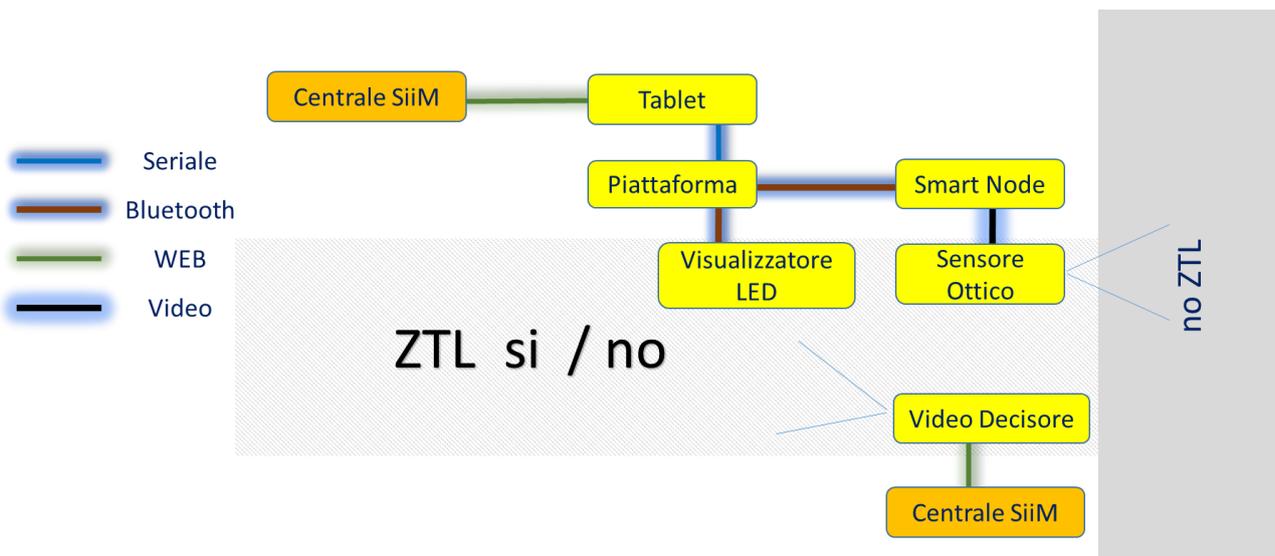


Figura 5 : Scenario di utilizzo del Segnale Dinamico

Il Sensore Ottico identifica il veicolo (leggendo la targa) e questa viene mandata alla Centrale Sii-Mobility via Tablet. A seconda che il veicolo sia autorizzato o meno, il Visualizzatore LED presenta informazioni diverse. Se non è autorizzato allora il Video Decisore manda un’immagine a Sii-Mobility per la sanzione.

Esiste una opzione particolarmente innovativa che consiste nell’identificare il veicolo facendo comunicare via Wi-Fi il Tablet con l’analogo dispositivo di bordo (KIT03 + APP02, si veda [SiiM 8]). In corso lavori si valuterà l’opportunità di realizzare questa funzione aggiuntiva.

Nel caso in cui il tratto di strada può essere escluso/incluso nella ZTL in modo dinamico, allora il Video Decisore serve anche a valutare se la strada è vuota per non sanzionare ingiustamente un veicolo che è già in transito al momento della commutazione. In questo caso la decisione potrebbe anche essere automatizzata dal momento che non ci sono implicazioni di sicurezza.

Da notare che il Video Decisore è completamente autonomo rispetto al Segnale Dinamico e si interfaccia direttamente con la Centrale Sii-Mobility.

3.3 Architettura generale

L’architettura generale, contestualizzata in Sii-Mobility, del Segnale Dinamico è mostrata nella *Figura 6* seguente.

3.4 Specifica di dettaglio

3.4.1 Diagramma di contesto generale

La *Figura 7* mostra un primo livello di specifica e ripropone lo schema di architettura nella notazione utilizzata per la descrizione di dettaglio. A questo livello occorre definire con maggior precisione le interfacce mostrate.

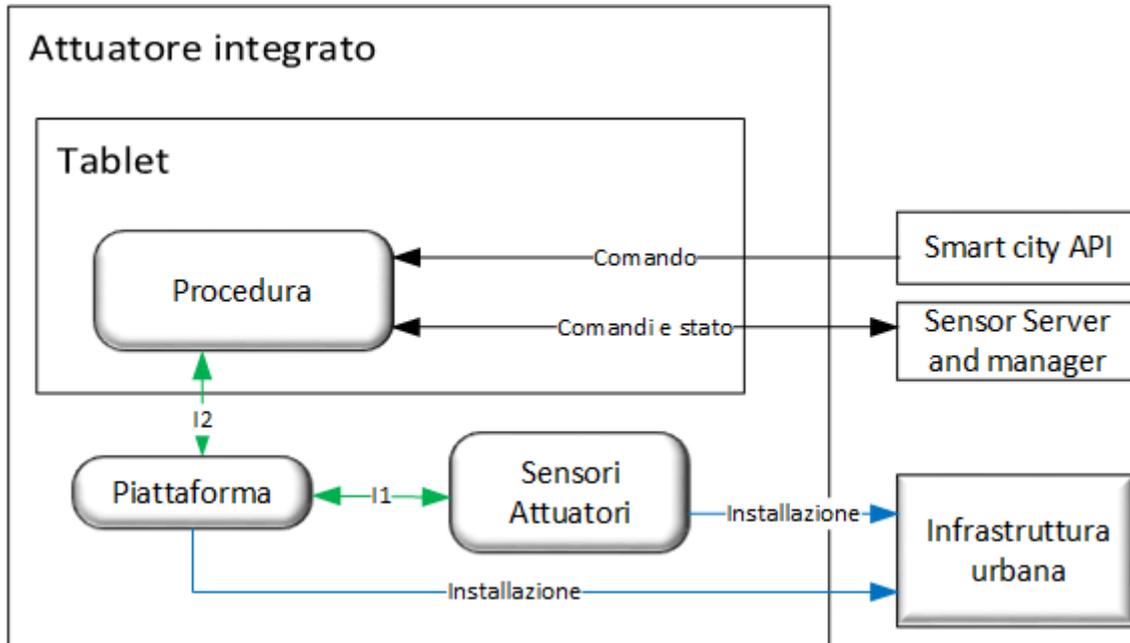


Figura 7 : Segnale Dinamico - diagramma di contesto generale

In particolare:

- Interfacce esterne:
 - *Comando* e *Comandi a stato* sono interfacce applicative realizzate con scambi di dati su WEB. Non vengono ulteriormente dettagliate in questo documento;
 - *Installazione* della Piattaforma e dei Sensori / Attuatori 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 e quelli che si occupano della installazione nel contesto urbano. L'argomento è sviluppato alla sezione 3.4.2.2;
- Interfacce interne:
 - **I1** è l'interfaccia fisica e logica tra piattaforma e sensori / attuatori. È sviluppata nelle sezioni che seguono;
 - **I2** è l'interfaccia fisica e logica tra procedura e piattaforma. È sviluppata nelle sezioni che seguono;

3.4.2 Piattaforma, EHW01

3.4.2.1 Diagramma di contesto

La *Figura 8* mostra 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 attivi potranno anche avere collegamenti di comando / controllo analoghi a quelli dei sensori passivi in aggiunta allo scambio di messaggi.

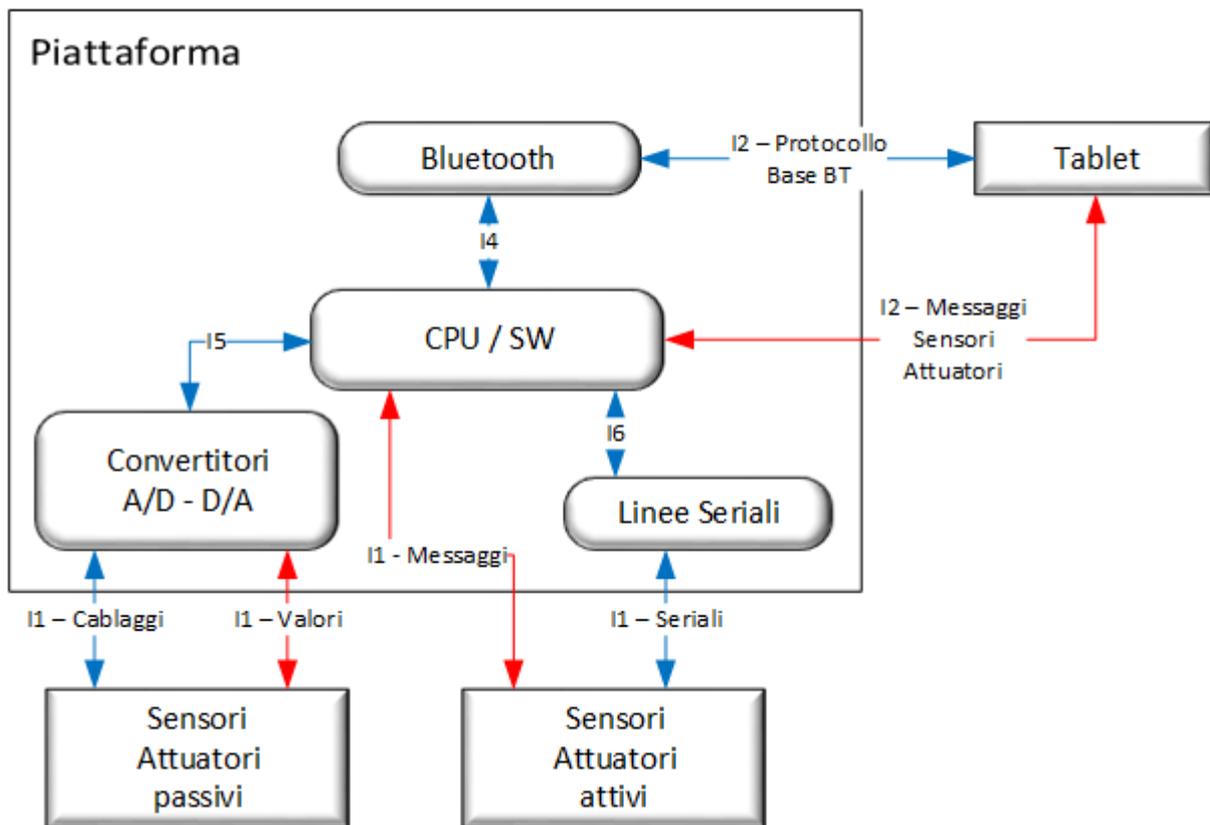


Figura 8 : Segnale Dinamico – diagramma di contesto della Piattaforma

La piattaforma comprende:

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

Principali attività e interfacce coinvolte:

- ❖ **gestione sensore / attuatore passivo** : *I1 – Cablaggi* rappresenta i collegamenti cablati con il sensore o attuatore mentre *I1 – Valori* rappresenta il significato della grandezza (tipicamente una tensione) che si acquisisce dal sensore / attuatore, o che si imposta per comandare l'attuatore. Secondo le convenzioni definite per *I2 – protocollo Base BT*, la piattaforma acquisisce i valori, li codifica in messaggi e li invia al tablet oppure decodifica i messaggi dal tablet (*I2 – Messaggi Sensori Attuatori*) e imposta i valori. È importante notare che viene trasmesso il dato grezzo del sensore, il suo significato (interfaccia logica) non è rilevante per la piattaforma, mentre lo è per la Procedura. Lo stesso vale per i comandi agli attuatori. L'interfaccia *I5* rappresenta le convenzioni, locali alla piattaforma (ECM-ELFI), sulla locazione dei convertitori e sulla loro gestione;
- ❖ **gestione sensore attivo** : i messaggi dai sensori / attuatori attivi (*I1 – Messaggi*) sono ritrasmessi senza modifiche al tablet e viceversa (*I2 – Messaggi Sensori Attuatori*) secondo le convenzioni definite per *I2 – protocollo Base BT*. L'interfaccia *I1 – Seriali* rappresenta solo il tipo di linea seriale, mentre *I6* 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 la Procedura (DISIT) che le recepisce;
- ❖ **gestione del collegamento Bluetooth** : il SW della piattaforma e quello del tablet definiscono e condividono le informazioni necessarie per realizzare un canale di comunicazione Bluetooth (interfaccia *I2 – protocollo Base BT*) con un suo protocollo. Utilizzano, rispettivamente, i servizi della piattaforma, interfaccia *I4*, e di Android (vedere *Figura 11*). Il protocollo comprende il formato dei messaggi (header, footer, etc.), messaggi di controllo e messaggi di dati. Nei messaggi dati sono contenuti (senza modifiche) i messaggi da e per sensori attivi e inoltre i messaggi generati dalla piattaforma per i sensori passivi.

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

- *I1 – Valori*
- *I1 – Messaggi*

3.4.2.2 Meccanica, contenitori, carpenterie, installazione

Come meglio dettagliato alla sezione 3.4.4.1.4, si pianifica un unico contenitore, in grado di contenere il Visualizzatore LED, la Piattaforma, il Tablet di servizio e lo Smart Node. Le Telecamere Intelligenti (Sensore Ottico e Video Decisore) potranno essere posizionate all'interno dello stesso contenitore oppure esternamente, sul contenitore stesso o su altro idoneo supporto.

Al momento non è possibile sviluppare completamente l'aspetto installativo, che richiede analisi congiunta con i gestori del traffico, dei parcheggi e di ZTL per individuare soluzioni funzionali al progetto, in linea con leggi e regolamenti vigenti e compatibili con le locazioni in cui potrà essere installato il contenitore.

I dettagli della installazione del Segnale Dinamico sono specificati nel relativo manuale [SiiM 10] a cura di ARGOS.

3.4.2.3 Hardware

3.4.2.3.1.1 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.

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 2]).

3.4.2.3.1.2 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.

3.4.2.3.1.3 Interfacciamento dei sensori e attuatori

La motherboard interagisce con i sensori e attuatori descritti nelle sezioni seguenti, ovvero uno Smart Node (3.4.4.1.1) a sua volta connesso ad un Sensore Ottico (telecamera intelligente) (3.4.4.1.2) e un Visualizzatore LED (3.4.4.1.4).

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 9 e la Tabella 1 mostrano uno schema generale e il dettaglio del collegamento delle interfacce fisiche alle porte di Arduino 101.

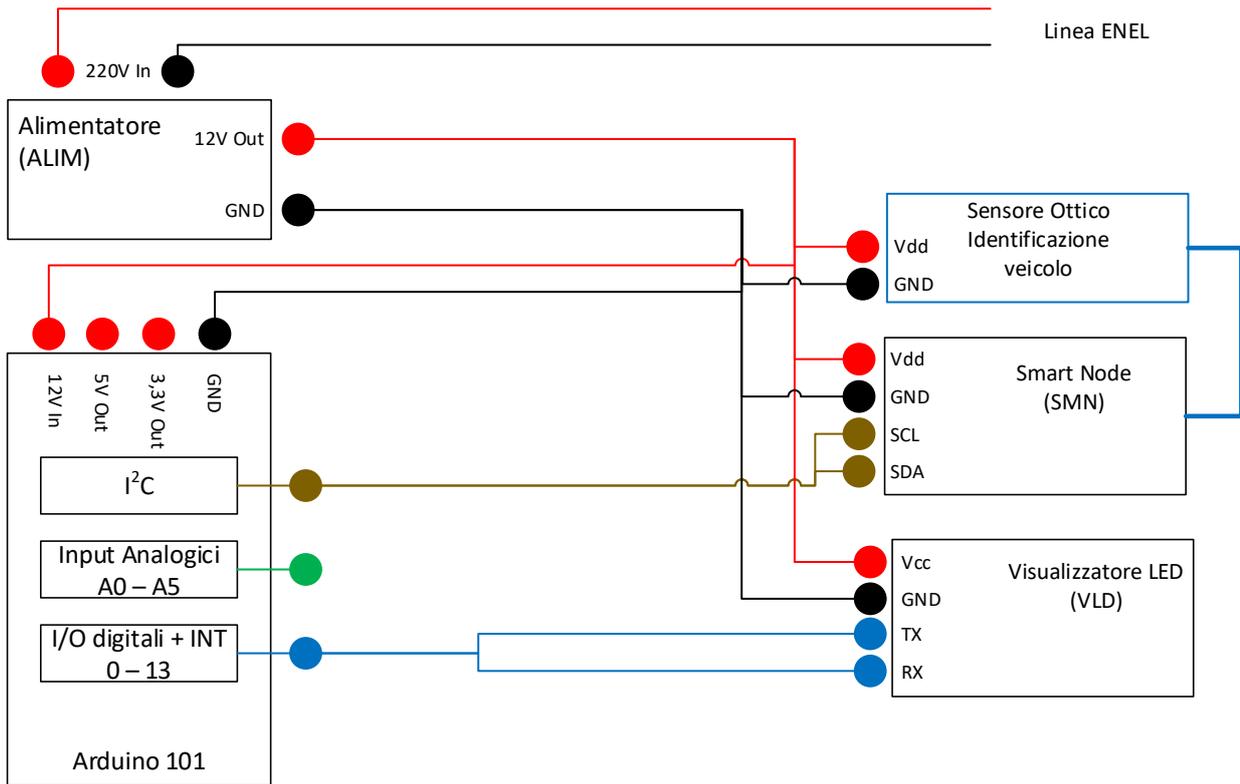


Figura 9 : Schema generale del Segnale Dinamico

Segnale Dinamico - Connessioni fisiche tra piattaforma e sensori							
Analog IN	A0	A1	A2	A3	A4	A5	
I/O/INT Ports	0/RX	1/TX	2°	3*	4	5**	6**
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 1: Segnale Dinamico - connessioni fisiche tra motherboard e sensori

*: porte che supportano PWM

°: porte che supportano interrupt sui cambiamenti di stato

3.4.2.4 Software

3.4.2.4.1 Casi d'uso

Nel Segnale Dinamico la Piattaforma è staticamente associata ad un Tablet di servizio ed inoltre per entrambi si prevede il funzionamento continuo. Di conseguenza il caso di uso corrispondente alla connessione dei due elementi via Bluetooth avviene sporadicamente e sempre tra gli stessi due attori.

Per il resto esiste un unico caso di uso corrispondente al funzionamento nominale, in cui la *Piattaforma* agisce come estensione della *Procedura* per la interazione con sensori e attuatori. Questo caso di uso è realizzato dai moduli SW definiti qui di seguito.

3.4.2.4.2 Architettura del software

La Figura 10 mostra la strutturazione di primo livello del software. Costituisce il dettaglio dell'elemento "CPU / SW" mostrato in Figura 8.

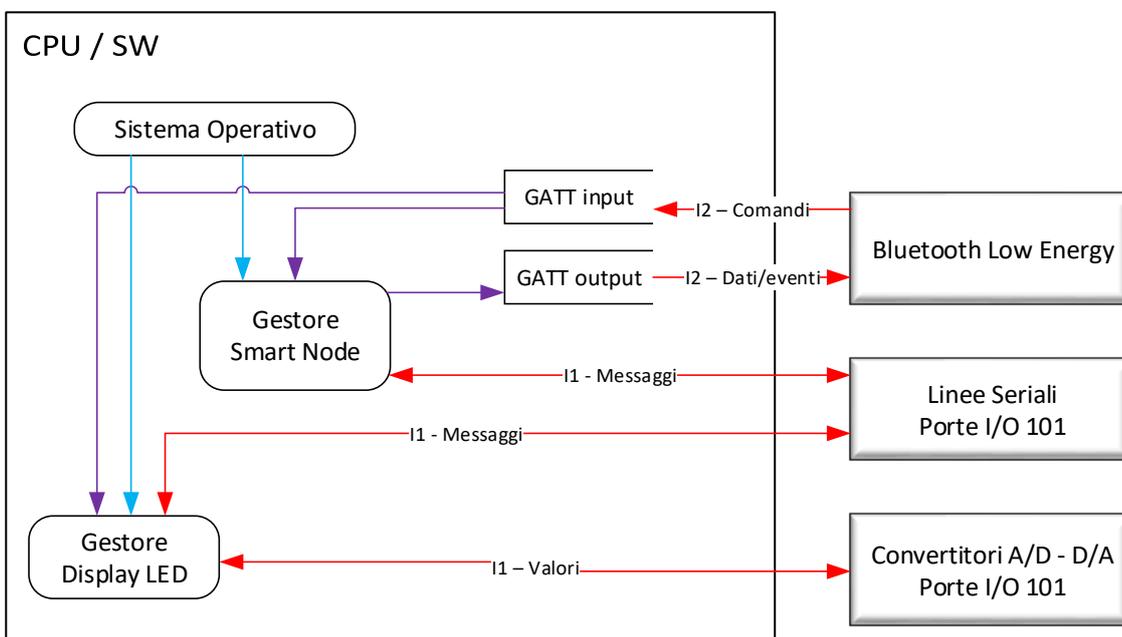


Figura 10 : Segnale Dinamico - architettura di base del software

Sono previsti due processi principali:

- **Gestore dello Smart Node** – riceve dallo Smart Node messaggi seriali contenenti l'identificazione del veicolo in avvicinamento. Inserisce il valore nell'opportuno GATT (General ATtribute Profile) per inviarlo alla *Procedura* residente sul *Tablet* di servizio. La *Procedura* può inviare comandi al processo per regolare il suo funzionamento mediante specifici GATT;
- **Gestore del Display LED** – riceve dalla *Procedura* comandi tramite appositi GATT in base ai quali regola l'aspetto del Display LED e l'informazione visualizzata. Al momento non sono previste comunicazioni dal processo alla *Procedura*. Si prevede la possibilità di utilizzare valori digitali per regolare parametri del Display, come ad esempio la luminosità.

La gestione dei processi è supportata da un opportuno sistema operativo, di tipo cooperativo, disponibile in modalità "open source".

3.4.3 Tablet di servizio

3.4.3.1 Diagramma di contesto

La *Figura 11* mostra il diagramma di contesto del Tablet di servizio.

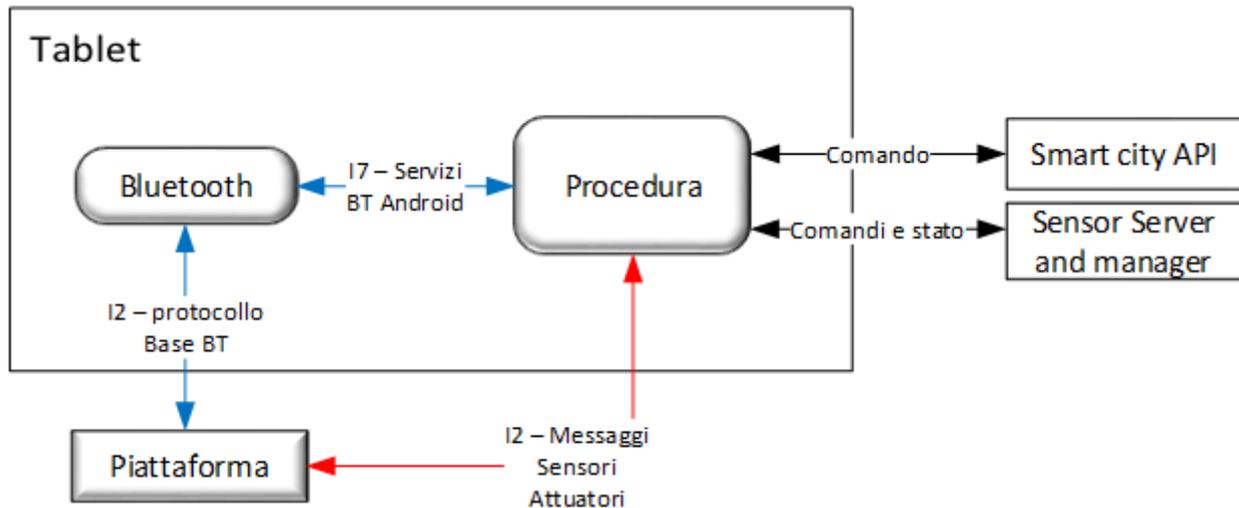


Figura 11 : Segnale Dinamico – diagramma di contesto del Tablet di servizio

Di seguito sono indicate le caratteristiche minime che il Tablet di servizio deve possedere per supportare efficacemente i Processi che vi sono installati:

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

3.4.3.2 APP con Procedura di Segnaletica, APP04

La procedura di segnaletica potrà prevedere di avere a disposizione delle risorse visive (immagini) da poter integrare all'interno del messaggio visuale mostrato e può avere dei messaggi preconfigurati e/o può decidere se visualizzare dei messaggi che arrivano dalla piattaforma. In modo da rendere tali messaggi dinamici e flessibili in base ai dati che devono essere visualizzati. Tali messaggi saranno salvati attraverso dei file JSON o XML data la loro flessibilità di utilizzo.

3.4.4 Sensori / Attuatori

Complessivamente, allo stato attuale, sensori e attuatori utilizzati sono:

- *Telecamera Intelligente*, sensore di tipo attivo costituito da uno Smart Node (SN04) collegato ad un Sensore Ottico (SN05). Complessivamente essi sono in grado di identificare un veicolo in avvicinamento leggendone la targa. Questo dato viene poi fornito dallo Smart Node alla Piattaforma tramite collegamento seriale. Entrambi gli elementi sono realizzati nel contesto dell'Attività 3.2;
- *Visualizzatore LED*, attuatore di tipo attivo in grado di visualizzare simboli e testo. È realizzato nel contesto dell'Attività 3.3 e la sua specifica di dettaglio è sviluppata in questo documento, alla sezione 3.4.4.1.4;
- *Sensori e attuatori* utilizzati solo per la Rastrelliera Intelligente. Sono realizzati nel contesto dell'Attività 3.3 e descritti alla sezione 5.4.3.

Il *Video Decisore* è invece connesso alla centrale in modo indipendente, invia immagini e riceve comandi in modo autonomo rispetto al Segnale Dinamico ed è citato solo per completezza.

In corso progetto sarà possibile pianificare il collegamento di altri sensori / attuatori resi disponibili dall'Attività 3.2 oppure internamente all'Attività 3.3, allo scopo di supportare nuove funzioni oppure migliorare l'implementazione di quelle già previste. Si tenga comunque presente che non è consentito utilizzare barriere mobili nel contesto delle porte ZTL in quanto potrebbero creare intralcio al transito dei mezzi di soccorso;

3.4.4.1.1 Smart Node, SN04

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

Per il Segnale Dinamico, l'informazione complessivamente fornita è il numero di targa del veicolo in avvicinamento.

Questo apparato è descritto nel deliverable dedicato ai sensori innovativi ([SiiM 4]/[SiiM 11]) alla sezione 5.

3.4.4.1.2 Sensore Ottico, SN05

Il Sensore Ottico è una telecamera (non necessariamente nel visibile) intelligente con capacità di elaborazione dell'immagine diversa a seconda della funzione che realizza. È sempre connesso a uno Smart Node con cui condivide il carico di elaborazione dell'immagine

Per il Segnale Dinamico, l'informazione complessivamente fornita è il numero di targa del veicolo in avvicinamento.

Questo apparato è descritto nel deliverable dedicato ai sensori innovativi ([SiiM 4]/[SiiM 11]) alla sezione 3.1.

3.4.4.1.3 Video Decisore, SN07+EHW02

Il *Video Decisore* è connesso alla centrale in modo indipendente, invia immagini e riceve comandi in modo autonomo rispetto al Segnale Dinamico ed è citato solo per completezza.

Il *Video Decisore* è dettagliato in [SiiM 4]/[SiiM 11] alla sezione 4.1.

3.4.4.1.4 Visualizzatore LED

Come dettagliato nella prima revisione di questo documento ([SiiM 5]), in una prima fase si era considerata la possibilità di sviluppare un visualizzatore di nuovo tipo, ispirato ai pannelli pubblicitari, dotato di uno schermo LED con particolari accorgimenti per massimizzare luminosità e contrasto e di un controller video in grado di generare pagine HTML su comando della Piattaforma.

Per questa soluzione è stato sviluppato, dal partner ARGOS, un disegno di dettaglio molto dettagliato ed accurato, in due fasi successive documentate in [SiiM 13].

Questo lavoro ha permesso di tener conto di tutte le difficoltà per ottenere la necessaria protezione ambientale e di stimare con precisione i costi di produzione. Tenendo anche conto della efficacia tutta da dimostrare dei meccanismi per ottenere una visibilità soddisfacente in pieno sole ma anche e soprattutto delle verosimili difficoltà ad ottenere da una PA il permesso di installare prototipi senza alcuna omologazione nel contesto cittadino si è deciso un approccio alternativo.

Si utilizza quindi un prodotto commerciale, di una ditta toscana leader nel settore dei pannelli indicatori stradali, realizzato appositamente per le porte ZTL. È fornito delle omologazioni che attestano la sua piena rispondenza a norme e regolamenti e la sua utilizzabilità nel contesto stradale.

Il visualizzatore LED è un pannello a matrice di LED, mostrato nella figura qui a lato, capace di visualizzare testo e simboli, specifico per i segnali stradali, conforme dal punto di vista normativo. Trattandosi di un segnale stradale è infatti necessario che vengano garantite le condizioni di luminosità e contrasto, e la grandezza del segnale, che permettano una buona visualizzazione da parte degli automobilisti, e che siano garantiti materiali e realizzazione resistenti agli agenti atmosferici. Un prodotto conforme dal punto di vista della sicurezza stradale garantisce anche la possibilità di un effettivo utilizzo “in campo” da parte delle amministrazioni e di chi si occupa della viabilità. Dall’indagine di mercato effettuata questo tipo di visualizzatore risulta sostenibile anche sul piano dei costi.

Dispone di tre righe grafiche (per indicazione di data, orario e condizione del varco) e un’area specifica per la visualizzazione del segnale di aperto (freccia verde) o chiuso (croce rossa).



Il pannello ha dimensioni 80 x 110 x 15 cm circa; è previsto staffaggio su un palo (9 cm di diametro e altezza 3 metri). È dotato di collegamenti RS485/Ethernet per lo scambio di dati.

Si pianifica di modificare il contenitore, mantenendone comunque le caratteristiche di protezione, per inserire la Piattaforma e il Tablet di servizio al suo interno e dotarlo di uno sportello che consenta di operare sul tablet

3.4.5 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 8* e *Figura 11*.

La specifica deve essere condivisa tra chi sviluppa il software della Piattaforma e della Procedura, 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.

Ci si propone di sviluppare il protocollo seguendo lo schema di comunicazione “nativo” Bluetooth Low Energy, descritto in 3.4.2.3.1.2. 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.

4 Direzioneatore

4.1 Struttura

Analogamente al Segnale Dinamico, il Direzioneatore 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* (e inoltre ne può contenere altri).

Altro componente fondamentale è una APP che realizza una *Procedura* in grado da un lato di comunicare con la centrale Sii-Mobility per scambiare dati, comandi e notifiche e dall'altro di interfacciarsi con la Piattaforma e quindi con Sensori e Attuatori.

Per lo sviluppo e il supporto della *Procedura* si utilizza un *Tablet* di servizio, in modo da fornire un host espressamente pensato per applicazioni WEB based. Il Tablet è componente sostanziale del Direzioneatore.

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

Si veda la sezione 3.1 per altri dettagli su questa organizzazione e relativi vantaggi.

4.2 Scenario di utilizzo

Il tipico scenario di utilizzo del Direzioneatore è mostrato nella *Figura 12*:

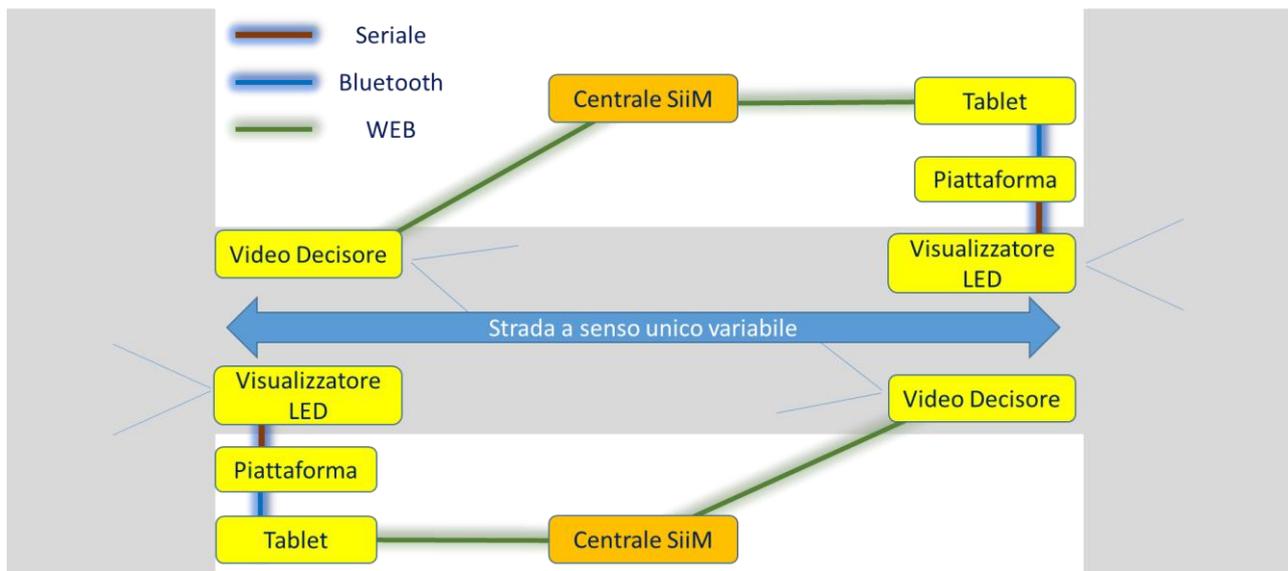


Figura 12 : Scenario di utilizzo del Direzioneatore

I Visualizzatori LED a regime hanno sempre aspetto complementare – libero accesso a un lato e divieto di accesso dall'altro. Le commutazioni vengono comandate dalla centrale Sii-Mobility in una sequenza coordinata.

Quando si commuta occorre verificare che la strada sia vuota: ne va la sicurezza delle persone, quindi non può decidere il software. Allora si predispose il divieto di accesso da entrambi i lati, poi il Video Decisore manda immagini a Sii-Mobility e un operatore umano deve dare il consenso a predisporre il libero accesso ad un estremo della strada.

Occorrono sempre (almeno) due Direzionatori, mentre a seconda della geometria della strada può bastare un solo Video Decisore.

Da notare che il Video Decisore è completamente autonomo rispetto al Segnale Dinamico e si interfaccia direttamente con la Centrale Sii-Mobility. Invia immagini e riceve comandi in modo autonomo rispetto al Segnale Dinamico e, quindi, non sarà ulteriormente descritto nel seguito di questo documento. Il *Video Decisore* è dettagliato in [SiiM 4]/[SiiM 11] alla sezione 4.1.

4.3 Architettura generale

L'architettura generale, contestualizzata in Sii-Mobility, del Direzionatore è mostrata nella seguente *Figura 13*.

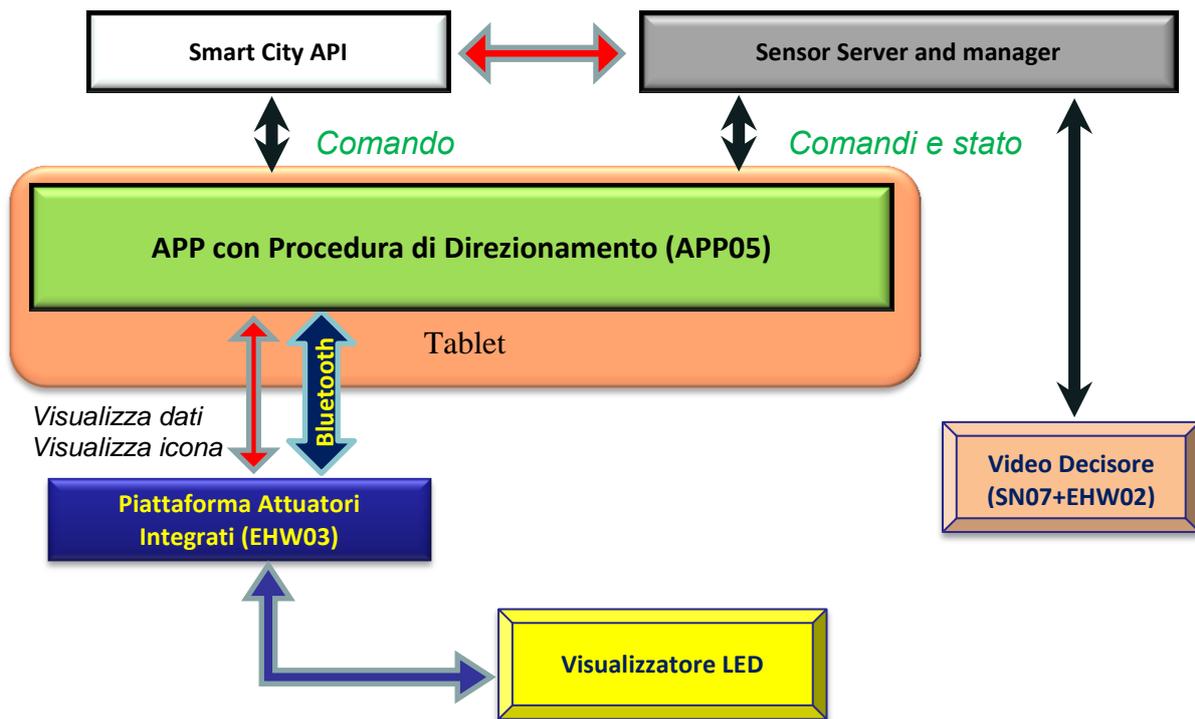


Figura 13 : Architettura generale del Direzionatore

Il Direzionatore è progettato per essere utilizzato nel contesto della modifica dinamica del senso di percorrenza di una strada o corsia. Ha la funzione di presentare icone / scritte che forniscono informazioni di libero accesso oppure di accesso vietato / limitato mediante l'attuatore *Visualizzatore LED*.

Chiaramente, la decisione di invertire il senso unico di una strada può essere attuata in sicurezza solo se in quel momento non ci sono veicoli in transito. Per questo motivo, un *Video Decisore* viene utilizzato per rilevare lo stato di "strada libera".

Il *Video Decisore* **non** è collegato direttamente all'attuatore integrato ma alla centrale Sii-Mobility per motivi di sicurezza. Infatti, non è possibile affidarsi al giudizio di un dispositivo elettronico intelligente per acquisire un'informazione che, qualora errata, ha un impatto diretto sulla sicurezza delle persone. L'informazione deve essere necessariamente confermata da un operatore umano.

Per questi motivi, il *Video Decisore* dialoga direttamente con la centrale, trasmettendo anche un'informazione visiva in base alla quale l'operatore può confermare e autorizzare la

commutazione. Il *Video Decisore* non fa quindi parte dell'attuatore integrato e viene mostrato solo per completezza.

La sequenza operativa di azioni complessiva, realizzata dalla centrale Sii-Mobility, consiste in:

- impostazione di entrambi gli attuatori posti agli estremi della strada in stato di “accesso vietato”,
- attendere conferma della avvenuta commutazione,
- attendere l'informazione di “strada libera” dalla telecamera,
- chiedere conferma da parte dell'operatore,
- commutare allo stato “accesso consentito” uno dei due attuatori integrati.

Il ruolo della *Procedura* è ricevere le richieste di cambio di stato dell'attuatore integrato, attuarle, verificare l'attuazione e fornire conferma alla centrale.

4.4 Specifica di dettaglio

4.4.1 Diagramma di contesto generale

Il primo livello di specifica e lo schema di contesto generale del Direzionatore è del tutto analogo a quello del Segnale Dinamico, si rimanda quindi alla sezione 3.4.1.

4.4.2 Piattaforma, EHW03

4.4.2.1 Diagramma di contesto

Anche per quanto riguarda il contesto operativo della Piattaforma, vale per il Direzionatore quanto già specificato per il Segnale Dinamico, si rimanda quindi alla sezione 3.4.2.1.

4.4.2.2 Meccanica, contenitori, carpenterie, installazione

Si utilizza lo stesso apparato prodotto commerciale (che sarà modificato per integrare la Piattaforma e il Tablet) utilizzato per il Segnale Dinamico. Si veda la sezione 3.4.2.2.

I dettagli della installazione del Direzionatore sono specificati nel relativo manuale [SiiM 10].

4.4.2.3 Hardware

L'hardware della Piattaforma Direzionatore è del tutto analogo a quello del Segnale Dinamico, si rimanda quindi alla sezione 3.4.2.3 per tutto quanto riguarda la Motherboard e il Bluetooth Low Energy.

La motherboard interagisce unicamente con il Visualizzatore LED, descritto alla sezione 3.4.4.1.4.

La *Figura 9* e la *Tabella 1* alla sezione 3.4.2.3.1.3 valgono anche per il caso del Direzionatore per descrivere le connessioni con il Visualizzatore LED, mentre lo Smart Node e il Sensore Ottico non sono presenti.

4.4.2.4 Software

La struttura del software della Piattaforma del Direzionatore è del tutto analogo a quella del Segnale Dinamico, si rimanda quindi alla sezione 3.4.2.4, in cui la *Figura 10* rimane valida tenendo presente che è presente il solo processo **Gestore Display LED**.

4.4.3 Tablet di servizio

4.4.3.1 Diagramma di contesto

Per quanto riguarda il contesto operativo del Tablet di servizio, vale per il Direzionatore quanto già specificato per il Segnale Dinamico, si rimanda quindi alla sezione 3.4.3.1 e alla *Figura 11*.

4.4.3.2 APP con Procedura di Direzioneamento, APP05

La procedura di Direzioneamento deve avere i seguenti requisiti:

- Riuscire a cambiare i segnali presenti negli attuatori ai limiti della strada
- Attendere una conferma del cambiamento avvenuto (che entrambi i segnali indichino “accesso vietato”)
- Attendere una conferma di strada libera dalla telecamera
- Una volta che ha le due conferme arrivate dai segnali e dalla telecamera deve attendere il comando che proviene dalla centrale operativa di SiiMobility
- Una volta che il comando viene ricevuto vengono cambiati i segnali presenti negli attuatori

La procedura si rimette in ascolto di ulteriori modifiche da fare a livello di segnalamento.

4.4.4 Sensori / attuatori

Il Direzionatore utilizza unicamente il Visualizzatore LED, descritto alla sezione 3.4.4.1.4.

Il *Video Decisore* è invece connesso alla centrale in modo indipendente, invia immagini e riceve comandi in modo autonomo rispetto al Direzionatore ed è citato solo per completezza.

4.4.5 Protocollo base Bluetooth

Si utilizza lo stesso protocollo specificato per il Segnale Dinamico alla sezione 3.4.5.

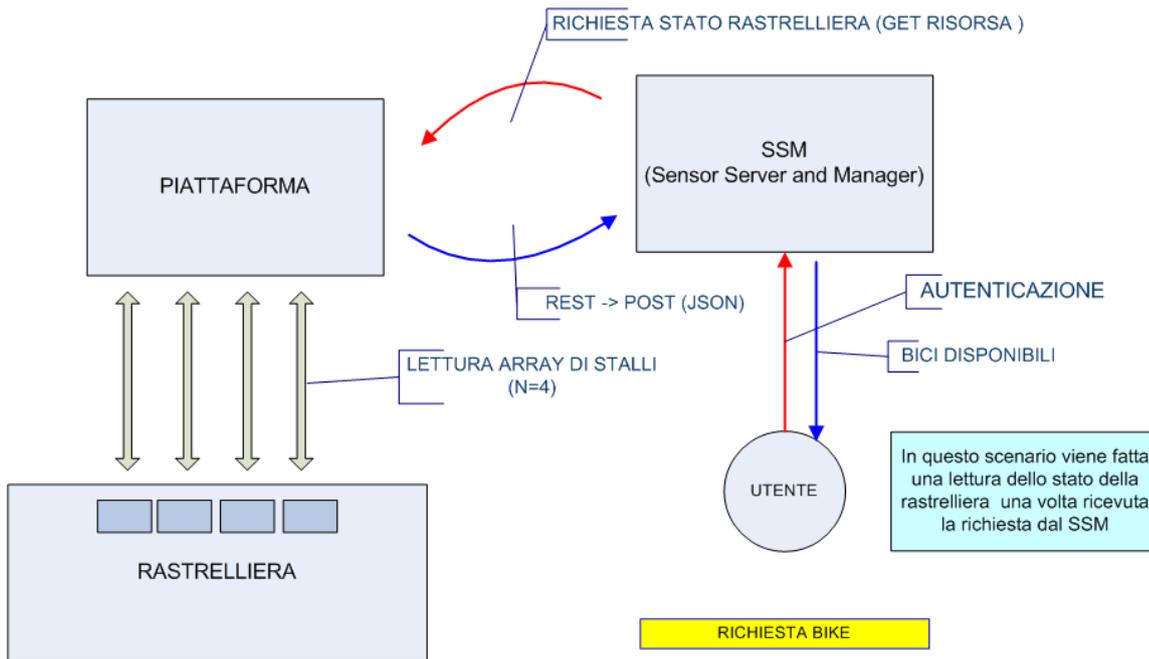


Figura 15 : Scenario 1- Utente viene aggiornato dello stato della rastrelliera tramite interrogazione del SSM

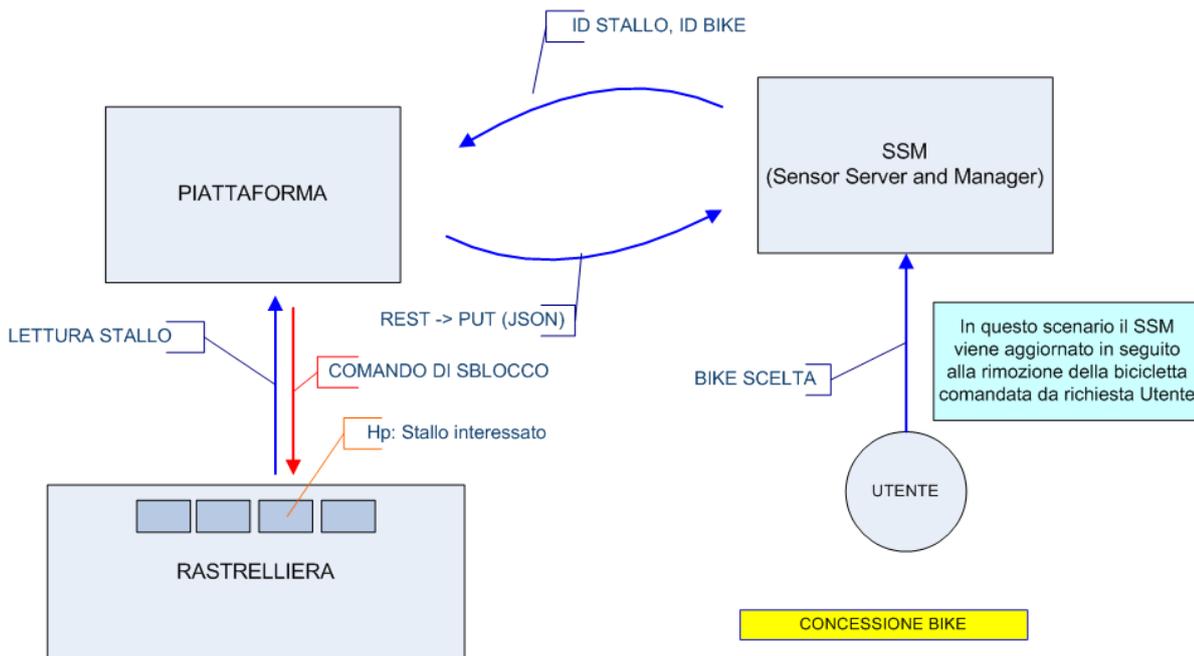


Figura 16 : Scenario 2- Trasmissione della richiesta di una bike e rilascio di questa a seguito di comando di sblocco da piattaforma

Figura 15 e Figura 16 descrivono in linea di principio il traffico informativo a partire dall'istante di accesso alla rastrelliera: dal momento in cui l'Utente accede alla rastrelliera (Figura 15) il SSM manda una richiesta della risorsa rastrelliera (stato stalli, id bikes) alla quale la piattaforma risponde dopo aver interrogato i lettori RFID. Al seguito dell'aggiornamento del SSM l'Utente è in grado di selezionare la bicicletta desiderata e tale richiesta viene inoltrata alla piattaforma la quale, tramite comando di sblocco, comanda l'apertura del gancio meccanico (vedi 5.4.2.3.1.2) : lo stato dello stallo viene letto (stato libero) in modo tale da controllare l'effettiva avvenuta apertura del gancio

(tale funzionalità sarà inoltre garantita dalla lettura di sensori di finecorsa installati a bordo rastrelliera, vedi 5.4.3.2.1.1).

Come ultimo passo il SSM viene aggiornato dello stato della rastrelliera.

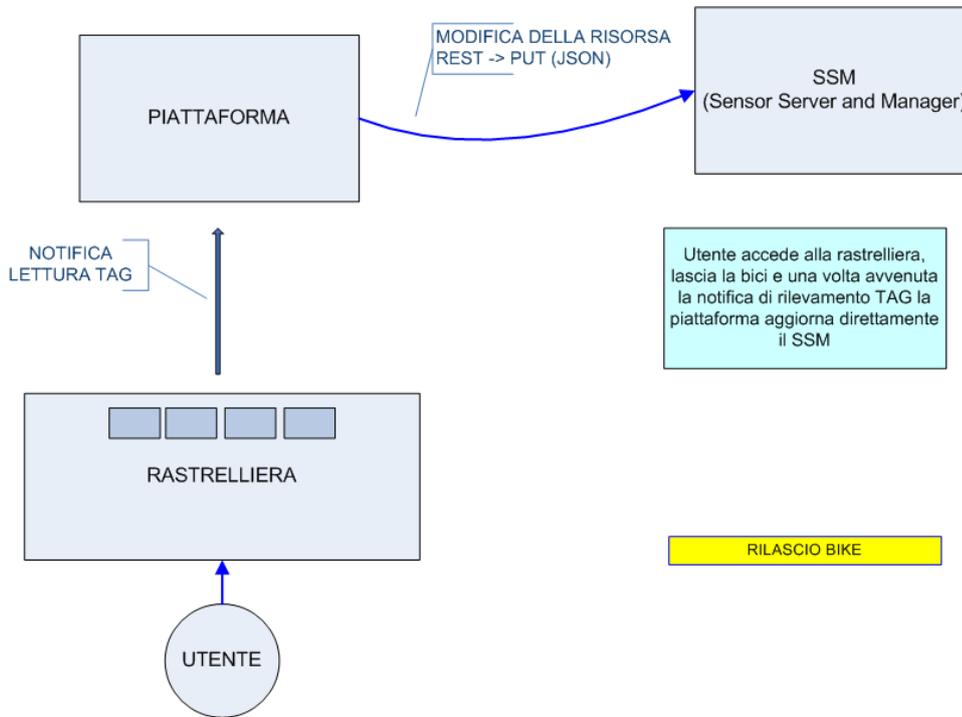


Figura 17 : Scenario 3- Al momento del rilascio della bike il sistema intelligente a bordo rastrelliera notifica la piattaforma la quale provvede all'aggiornamento dello stato comunicandolo al SSM

Figura 17 descrive lo scenario di rilascio della bike: in particolare, nel momento del rilascio, tramite interrupt (vedi 5.4.2.3.1.1) la piattaforma viene informata della rilevazione di un tag, acquisisce stato dello stallo e ID bike e aggiorna il SSM.

Gli scenari sopra descritti mettono in luce i flussi di informazione in sincronia con gli eventi di acquisizione e rilascio della bike: di base, poi, è previsto un aggiornamento periodico del server in merito allo stato della rastrelliera (messaggi JSON contenenti stato del singolo stallo, id bike, *timestamp*, vedi 5.4.2.3.1.1, 5.4.2.4.1).

5.3 Architettura generale

L'architettura generale, contestualizzata in Sii-Mobility, della Rastrelliera Intelligente è mostrata nella Figura 18 seguente.

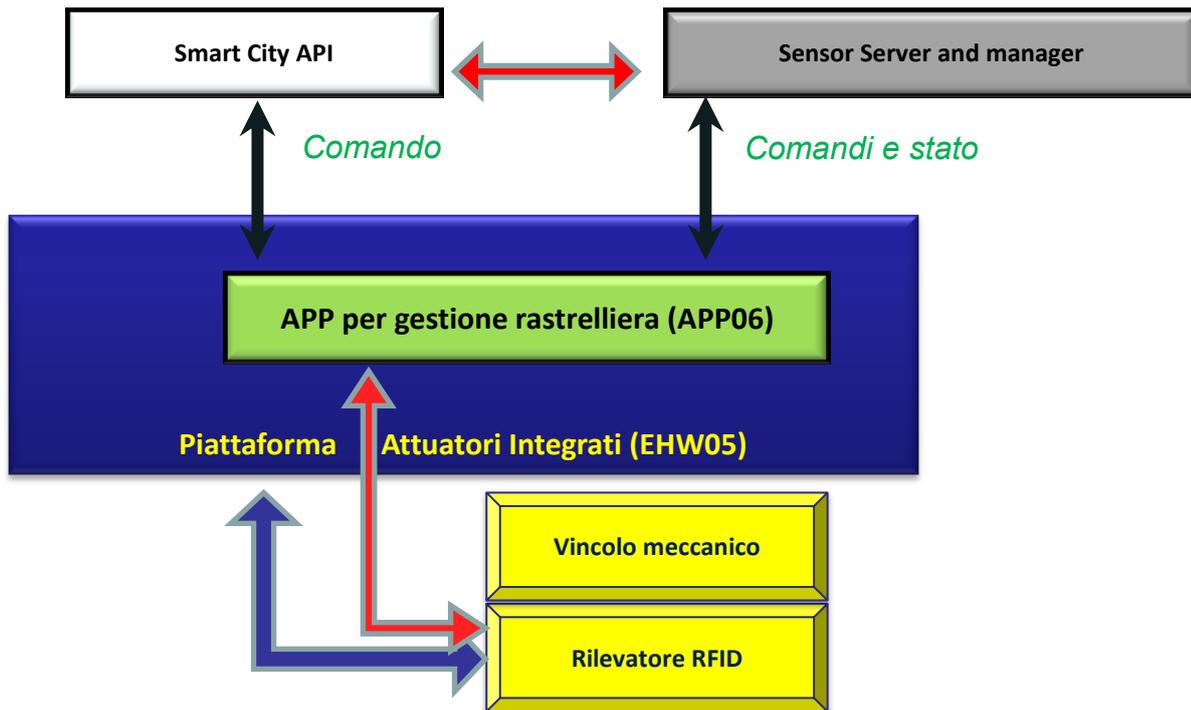


Figura 18 : Architettura generale della Rastrelliera Intelligente

La Rastrelliera Intelligente è progettata per realizzare la funzione **AATF** (autonomous anti-theft feature) principalmente o totalmente sull'infrastruttura di terra, aggiungendo elementi anche complessi all'infrastruttura ed eventuali elementi molto semplici sul kit bike.

La Rastrelliera Intelligente è in grado di comunicare autonomamente con la centrale Sii-Mobility.

Per quanto riguarda il meccanismo per rilevare o impedire il furto si hanno almeno due possibilità:

- i. il veicolo è equipaggiato con un *Rilevatore RFID* e la Piattaforma della Rastrelliera Intelligente è in grado di rilevarne la (mancata) presenza. Questo provoca un allarme a meno che dalla centrale Sii-Mobility non sia stata inviata un'autorizzazione, cosa che avviene quando la Mobile App dello Smartphone Utente ha completato l'associazione con il veicolo ed informa di questo la centrale;
- ii. la Rastrelliera Intelligente è dotata di *Vincoli Meccanici* che impediscono il prelievo del veicolo. Questi sono comandati e controllati dalla Piattaforma, che a sua volta è pilotata dalla Procedura del Tablet. Il comando di sblocco viene inviato dalla centrale Sii-Mobility alla Procedura quando la Mobile App dello Smartphone Utente ha completato l'associazione con il veicolo ed informa di questo la centrale. Quando l'Utente restituisce la bici, il vincolo meccanico viene ripristinato (occorre verificare il buon esito dell'operazione) e solo allora si chiude l'associazione con l'Utente.

In prima analisi la funzione antifurto verrà implementata secondo quanto descritto nel punto i) e, di conseguenza, l'intelligenza per la gestione di tale condizione è a carico della piattaforma, che dovrà implementare azioni di controllo in sinergia con la centrale Sii-Mobility (vedi [Figura 19](#)).

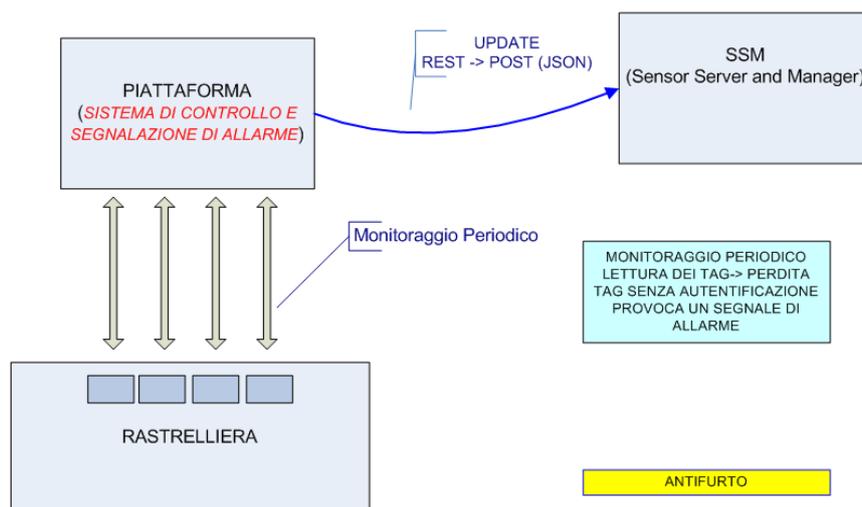


Figura 19 : Attraverso la lettura periodica dei tag è possibile rilevare la mancata presenza di una bike che, se non preceduta da una autorizzazione da parte del SSM, innesca il meccanismo di allarme a bordo della piattaforma

5.4 Specifica di dettaglio

5.4.1 Diagramma di contesto generale

La [Figura 20](#) mostra un primo livello di specifica e ripropone lo schema di architettura nella notazione utilizzata per la descrizione di dettaglio. A questo livello occorre definire con maggior precisione le interfacce mostrate.

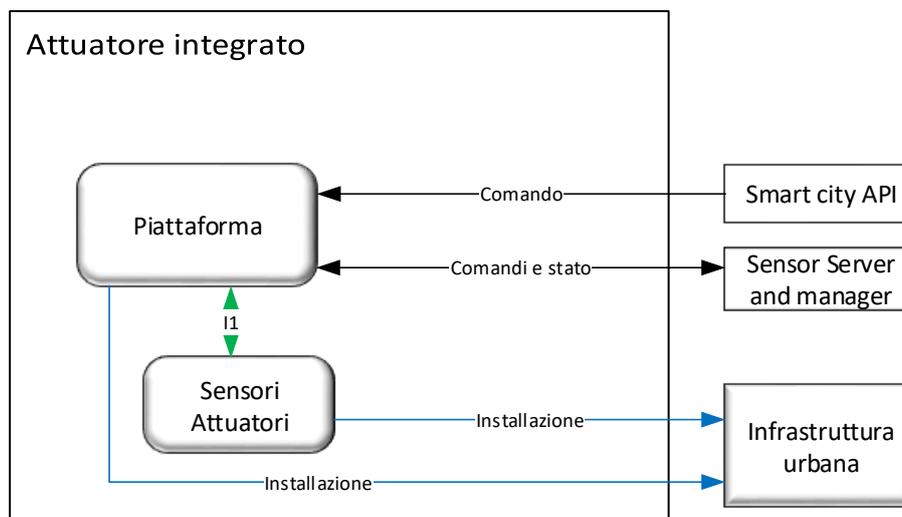


Figura 20 : Rastrelliera Intelligente - diagramma di contesto generale

In particolare:

- Interfacce esterne:
 - *Comando* e *Comandi a stato* sono interfacce applicative realizzate con scambi di dati su WEB. Non vengono ulteriormente dettagliate in questo documento;
 - *Installazione* della Piattaforma e dei Sensori / Attuatori 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 e quelli che si

occupano della installazione nel contesto urbano. L'argomento è sviluppato alla sezione 5.4.2.1;

- Interfacce interne:
 - I1 è l'interfaccia fisica e logica tra piattaforma e sensori / attuatori. È sviluppata nelle sezioni che seguono;

5.4.2 Piattaforma, EHW05

Nelle sezioni che seguono si dettaglia l'insieme dei moduli meccanici, elettromeccanici, hardware e software che vengono utilizzati per costruire la Rastrelliera Intelligente.

Sono inoltre specificati nel dettaglio sensori e attuatori utilizzati connessi alla Rastrelliera Intelligente.

5.4.2.1 Diagramma di contesto

La *Figura 21* mostra il diagramma di contesto della Piattaforma.

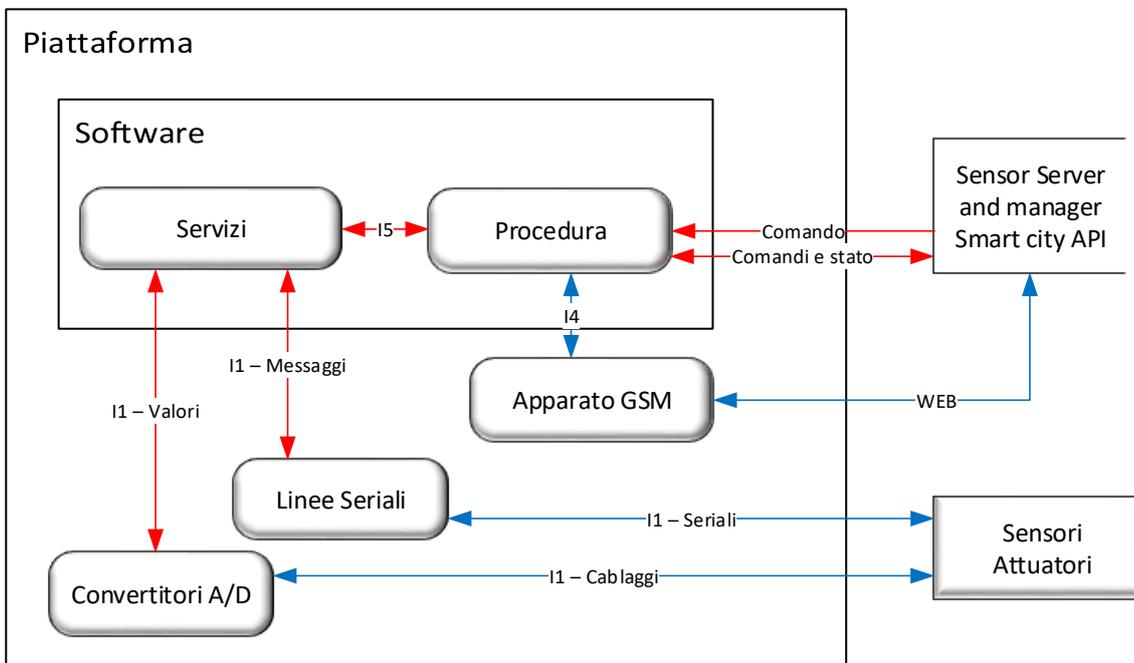


Figura 21 : Rastrelliera Intelligente - diagramma di contesto della Piattaforma

In particolare:

I1-Seriali: i dati relativi all'acquisizione dei tag della bike saranno trasmessi tramite interfaccia seriale I²C (vedi 5.4.3.1.1.2, Figura 24);

I1-Cablaggi:

- acquisizione tramite pin digitali dei segnali di interrupt generati dai singoli rilevatori RFID al momento della rilevazione di un tag (vedi 5.4.2.3.1.1);
- comando di relè per attuazione del gancio meccanico tramite livello alto del pin digitale (logica TTL 5V, vedi 5.4.3.2.1.1, Figura 35);

- acquisizione tramite pin analogici di finecorsa installati a bordo rastrelliera (vedi [Figura 39](#)).

II-Messaggi: la trasmissione segue la logica descritta in 5.4.3.1.1.2 e, nel caso in esame, i byte trasmessi riguardano il codice univoco del tag della bike. I livelli di tensione sono compatibili con quelli della piattaforma (5 V);

II-Valori:

- l'acquisizione tramite pin digitali (*interrupt da sensori RFID*) e la trasmissione dei comandi di pilotaggio relè segue la logica TTL (0-5 V);
- l'acquisizione tramite pin analogici è compatibile con i livelli logici operativi di Arduino UNO (vedi 5.4.3.2.1.1).

Per quanto riguarda la *I5* e la *I4* viene fatto riferimento a quanto specificato in 5.4.2.4.1, in cui sono elencati i campi che saranno parte della struttura dei messaggi JSON usati come scambio delle informazioni negli scenari descritti in 5.2, 5.3.

5.4.2.2 Meccanica, contenitori, carpenterie, installazione

Al momento non è possibile sviluppare operativamente questo aspetto, che richiede analisi congiunta con i gestori del traffico, dell'urbanistica e dei servizi di bike-sharing per individuare soluzioni funzionali al progetto, in linea con leggi e regolamenti vigenti e compatibili con le locazioni in cui potrà essere installata la Rastrelliera Intelligente.

Allo stato attuale, ARGOS ha prodotto una proposta di progetto di una rastrelliera con la prerogativa di essere flessibile a fronte di varie categorie di biciclette e dotata di un meccanismo di blocco/sblocco della bicicletta efficace ed efficiente. Una descrizione molto dettagliata ed accurata della (possibile) infrastruttura è contenuta nell'allegato [SiiM 14].

I dettagli della installazione della Rastrelliera Intelligente sono specificati nel relativo manuale [SiiM 10] a cura di ARGOS.

5.4.2.3 Hardware

5.4.2.3.1.1 Motherboard (D I E F)

La fase di scelta della motherboard si è sviluppata sulla base di alcuni requisiti:

- *Protocollo di comunicazione:* lo scambio di informazioni tra piattaforma e centrale Sii-mobility seguirà le convenzioni del protocollo HTTP, mediante chiamate REST e il formato dei messaggi sarà di tipo JSON;
- *Gestione dell'informazione:* non è prevista la presenza di un tablet, venendo così a mancare l'esigenza di una comunicazione Bluetooth;
- *Disponibilità di IN/OUT:* la piattaforma dovrà essere dotata di un numero di pins adeguato per acquisire i segnali provenienti dai lettori RFID e applicare i segnali di attuazione;
- *Infrastruttura della rastrelliera:* il numero di stalli previsto è pari a 4.

Sulla base dei requisiti suddetti la scelta è stata orientata verso la board Arduino UNO Rev.3: *Figura 22* illustra i collegamenti verso le periferiche quali lettori RFID, relè di comando degli attuatori elettromagnetici e acquisizione dei segnali provenienti dai finecorsa (vedi 5.4.3.2.1.1).

In particolare i collegamenti riguardano:

- comunicazione I^2C con la quale ogni stallo (lettore RFID) sarà interrogato al proprio indirizzo;
- lettura dei segnali interrupt di avvenuto rilevamento di un tag (vedi 5.4.3.1);
- segnali di controllo dei relè responsabili dell'azionamento elettromagnetico (vedi *Figura 25*);
- segnali di stato del gancio tramite lettura dei finecorsa (vedi *Figura 39*).

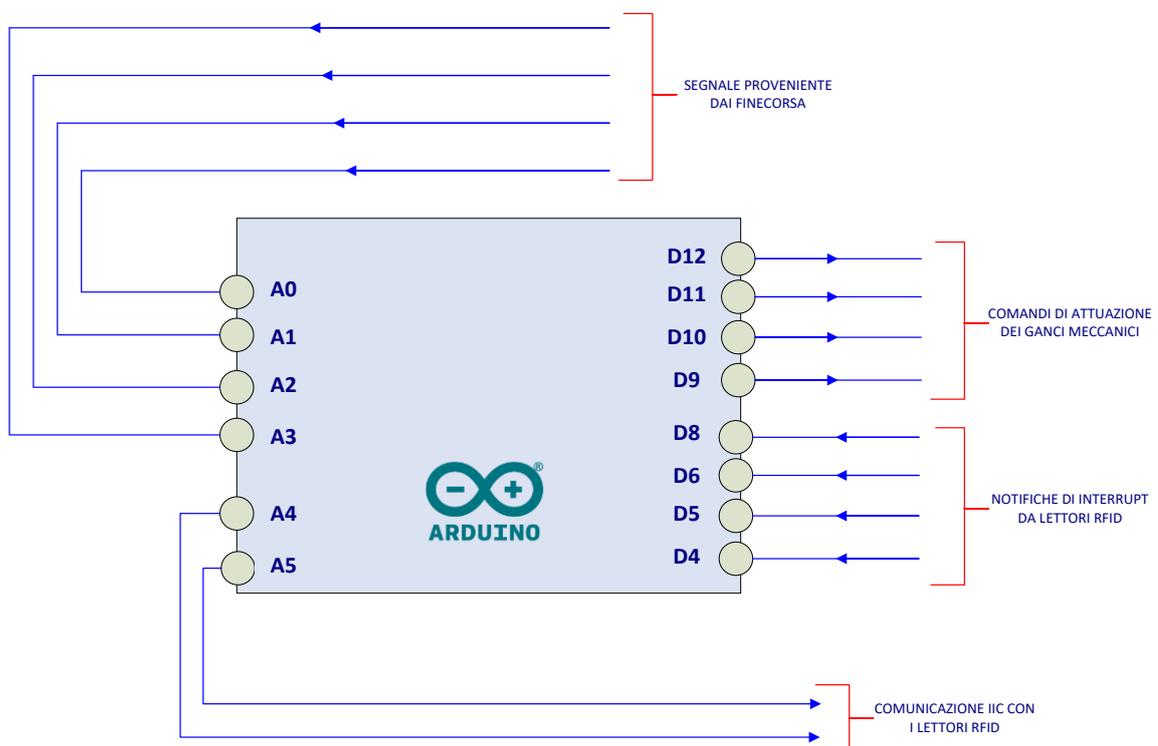


Figura 22 : Collegamenti tra la motherboard, sensori RFID (vedi 5.4.3.1) e attuatori (vedi 5.4.3.2.1.3)

Al fine di garantire lo scambio di informazioni tra la piattaforma e il server è stato adottato un protocollo di comunicazione che si appoggia su rete GSM/GPRS: a tal proposito è stata scelta una shield in grado di fornire le funzionalità minimali di scambio di messaggi JSON attraverso chiamate REST.

La Arduino GSM shield consente ad una scheda Arduino di connettersi ad internet, effettuare / ricevere chiamate vocali ed inviare / ricevere messaggi SMS.

La shield utilizza un modem radio M10 della Quectel. Le principali caratteristiche sono indicate in Tabella 2.

PARAMETRI	
Tensione Operativa	5 V
Connessioni	GSM e GPRS
Tipi di Comunicazione	SMS, chiamate,
Protocolli Supportati	TCP / UDP / HTTP attraverso connessione GPRS
Frequenze di funzionamento	GSM850MHz, GSM900MHz, DCS1800MHz e PCS1900MHz
Velocità massima di trasferimento dati GPRS	85,6 kbps
Antenna	Integrata
Connessioni con Arduino	Collegamento con pin 2(TX) e 3(RX) della scheda UNO(software seriale) e pin 7 (RESET)

Tabella 2: Principali caratteristiche del modulo shield GSM/GPRS

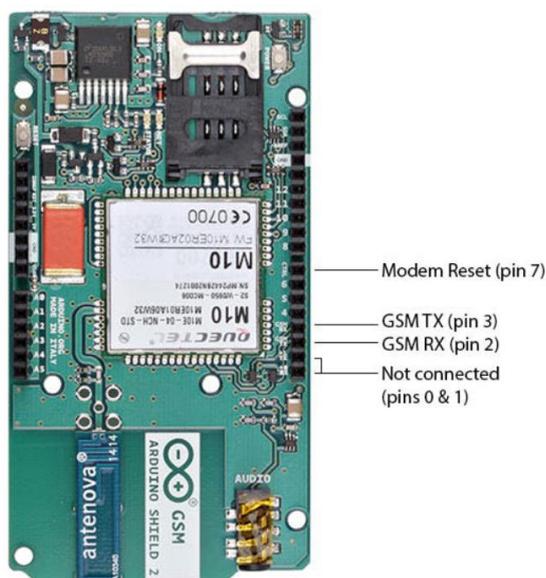


Figura 23 : Modulo Shield GSM/GPRS e connessioni con Arduino UNO

Il modulo GPRS sarà strettamente utilizzato per comunicare con la centrale Sii-Mobility attraverso il protocollo di livello applicativo HTTP.

Le applicazioni basate su REST, si definiscono **RESTful** e utilizzano le richieste HTTP per inviare i dati (creazione e/o aggiornamento), effettuare query, modificare e cancellare i dati. In definitiva, REST utilizza HTTP per tutte e quattro le operazioni CRUD (Create / Read / Update / Delete). Un servizio REST è:

- Indipendente dalla piattaforma
- Indipendente dalla lingua
- Basati su standard (HTTP in primis)
- Può essere facilmente utilizzato in presenza di firewall

Le risorse saranno rappresentate da messaggi JSON basati su:

- **Coppie nome/valore:** realizzato come un oggetto, un record, un dizionario, una tabella hash, un elenco di chiavi o un array associativo (per l'applicazione in oggetto vedi 5.4.2.4.1).

Per quanto riguarda il protocollo della comunicazione tra modulo GSM/GPRS e SSM (Sensor Server and Manager) si rimanda agli scenari descritti in 5.2.

5.4.2.3.1.2 Interfacciamento dei sensori e attuatori

La motherboard interagisce con i sensori / attuatori descritti nelle sezioni seguenti, ovvero Rilevatore RFID (5.4.3.1), e Vincolo Meccanico (5.4.3.2).

Le alimentazioni dei sensori / attuatori 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.

Di seguito viene definita l'architettura di sistema per il monitoraggio di N stalli della bike che avviene secondo protocollo di comunicazione I^2C (vedi [Figura 24](#)). In [Figura 25](#) invece è definito l'insieme di connessioni per il comando dell'attuatore lineare a solenoide che permette il gancio/sgancio della bike al singolo stallo: anche in questo caso lo schema è fatto per rappresentare uno scenario di N stalli che equipaggiano la rastrelliera. La rastrelliera sarà anche equipaggiata di sensori di finecorsa per avere un feedback sullo stato del gancio: la [Tabella 3](#) riassuntiva dei collegamenti è compilata prendendo in esame un solo stallo della rastrelliera e considera le funzionalità di comunicazione con i lettori RFID, comando dei relè per gli attuatori e lettura stato dei finecorsa (vedi 5.4.3.2.1.1).

Rastrelliera Intelligente - Connessioni fisiche tra piattaforma e sensori							
Analog IN	A0	A1	A2	A3	A4	A5	
	SIG _{fc}						
I/O/INT Ports	0/RX	1/TX	2°	3*	4	5**	6**
			PN532-D2.OUT	Relè-CTRL			
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
	PN532(SCL)	PN532(SDA)					

Tabella 3: Rastrelliera Intelligente - connessioni fisiche tra motherboard e sensori

*: porte che supportano PWM

°: porte che supportano interrupt sui cambiamenti di stato

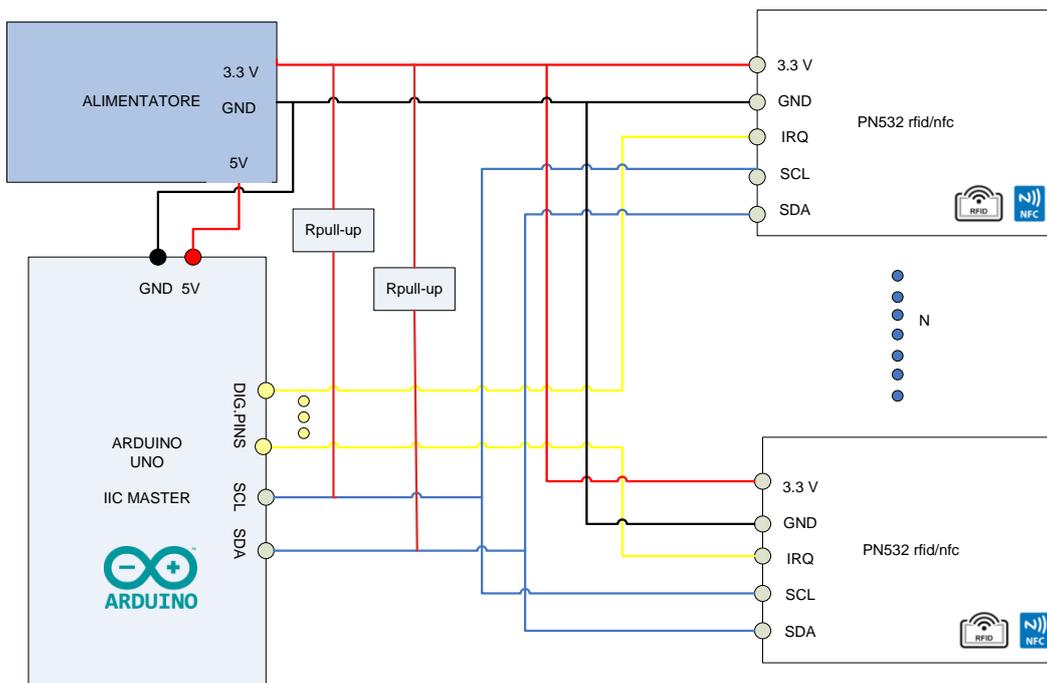


Figura 24 : Schema generale della Rastrelliera Intelligente-definizioni delle connessioni per la parte di raccolta dati proveniente dai sistemi RFID di ogni stallo

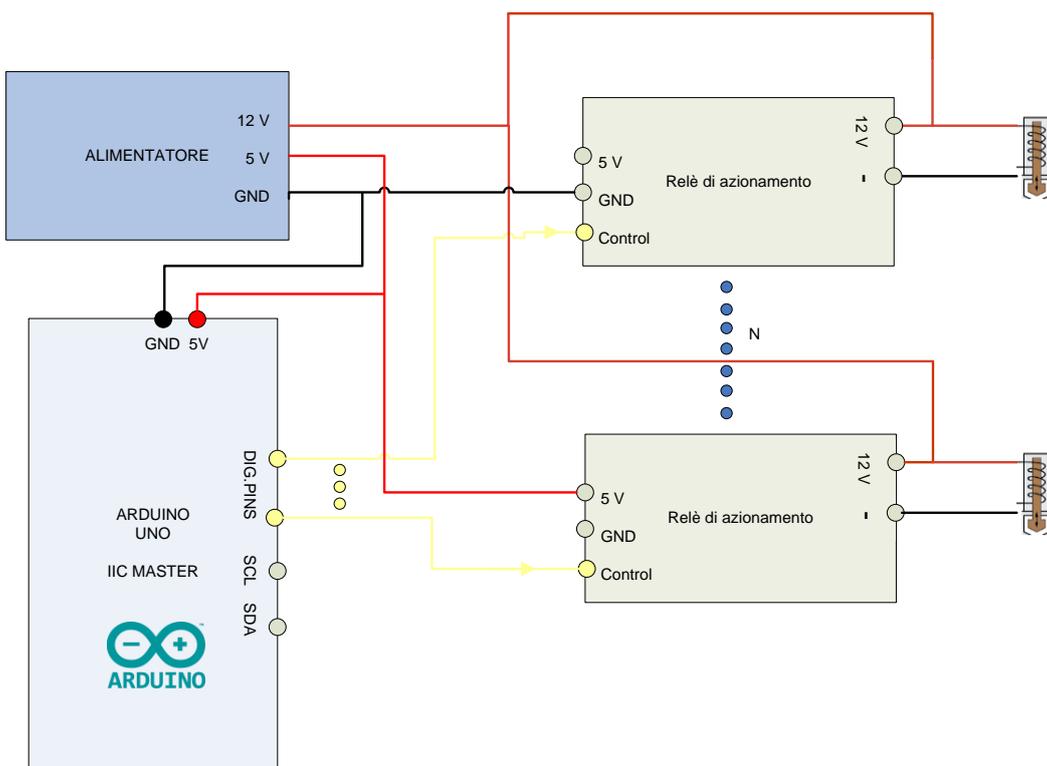


Figura 25 : Schema generale della Rastrelliera Intelligente-definizioni delle connessioni di attuazione dei solenoidi per aggancio/sgancio della bicicletta allo stallo

5.4.2.4 Software

Il Software dovrà implementare una comunicazione HTTP attraverso chiamate GET per gestire lo scambio di informazioni/dati tra piattaforma e rastrelliera.

In aggiunta, oltre alla comunicazione dei dati in sincronia con gli eventi di accesso (prelievo) e rilascio della bike, è previsto un aggiornamento periodico del SSM sullo stato della rastrelliera (servizi).

5.4.2.4.1 Servizi

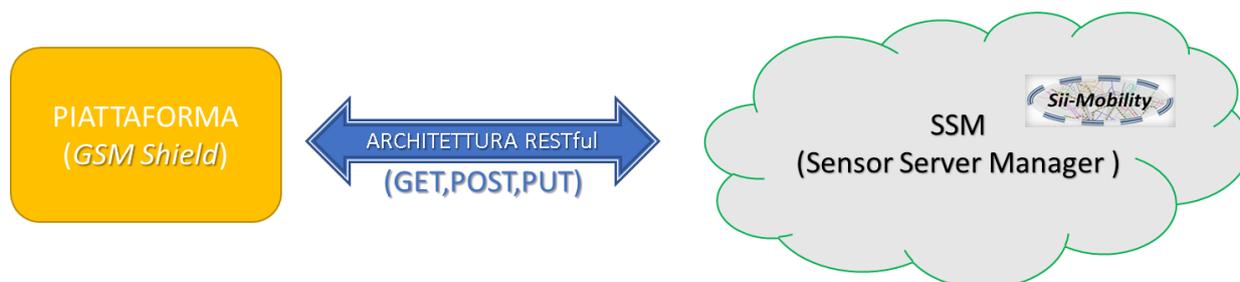


Figura 26 : Realizzazione di un Web Service secondo architettura in stile REST

Oggetto di studio sarà la definizione di un Web Service aderente all'architettura in stile REST (vedi 5.4.2.3) per la quale è richiesta la creazione di un client RESTful (piattaforma + GSM shield) attraverso la definizione di:

1. Client http;
2. URI delle risorse oggetto della comunicazione regolamentata da richieste di tipo GET, POST, PUT;
3. Definizione del tipo di rappresentazione delle risorse (JSON);
4. Capacità di interpretare il formato JSON.

Come detto in precedenza, le risorse saranno rappresentate attraverso messaggi JSON che contengono le informazioni di ogni stallo (in numero pari a N), in particolare:

- ID dello stallo (codice univoco rappresentato da un numero intero tra $1 \div N$);
- Stato dello stallo (numero intero che può assumere i valori 0/1/2);
- Tag Univoco della Bike;
- Timestamp: tempo di aggiornamento stato.

5.4.2.4.2 APP per gestione rastrelliera, APP06

L'applicazione dovrebbe avere le seguenti caratteristiche:

- Offrire in ogni istante la possibilità di prendere in affitto una bici, tramite il nostro servizio.
- APP disponibile per ogni sistema operativo mobile.
- Registrazione utente con la quale effettuare prenotazioni e pagamenti.
- Controllo in app GPS per permettere prenotazioni a rastrelliere vicine

In dettaglio gli obiettivi relativi al kit veicolare dovranno essere :

- Interfaccia facile ed intuitiva
- Localizzare rastrelliere vicine
- Ritiro e deposito di una bici in una determinata rastrelliera
- Registrazione:
 - richiesta di una bici (prenotazione in un raggio massimo prestabilito, per un periodo limitato)
- Pagamento effettuato tramite sistema di crediti prepagato
- Sistema di feedback

5.4.2.4.3 Progetto di dettaglio

Il progetto prevede i seguenti contesti:

LOCALIZZAZIONE RASTRELLIERE

L'APP, previa attivazione gps sul dispositivo, deve essere in grado di segnalare le rastrelliere più vicine, dicendo inoltre il numero di bici disponibili nei rispettivi slot.

Inoltre fornisce informazione sulla disponibilità di slot liberi per il deposito.

Per usufruire del servizio di ritiro o consegna bisogna trovarsi ad una distanza (5/10 metri) predefinita dalla rastrelliera.

REGISTRAZIONE UTENTE

Sarà disponibile una pagina di registrazione nella quale saranno richiesti dati come: nome cognome, data di nascita, numero di telefono e mail (devono essere verificate per permettere la registrazione)

RITIRO BICI

- L'APP si collega con la rastrelliera che a sua volta è collegata alla rete.
Inizia il processo di dialogo tra rastrelliera app e server.
- La rastrelliera sa in ogni istante quale dei suoi slot è libero o occupato.
- L'utente registrato e collegato alla rastrelliera manda la richiesta di ritirare una bici.
- Il server riconosce se l'utente è registrato.
 1. Se è registrato procede al passo successivo
 2. Se non è registrato viene rimandato alla procedura di registrazione
- L'utente viene riconosciuto.
- Il server manda l'ACK
- La rastrelliera sblocca una bici per un periodo di tempo, in quel periodo l'utente è tenuto a ritirare la bici.

- La bici ha un tag e la rastrelliera un lettore, il codice tag viene quindi associato all'utente.
- Lo slot dove era precedentemente inserita la bici rimarrà aperto per una durata predefinita (5m) in quel tempo l'utente può riinserire la bici. Passati il tempo lo slot diventerà vuoto e sarà disponibile anche ad altri utenti per il deposito

CONSEGNA BICI

- Quando l'utente ha finito di utilizzare una bici e vuole procedere al deposito si collega tramite l'app (previa attivazione gps) per localizzare le rastrelliere più vicine.
- L'app mostra all'utente le rastrelliere nelle vicinanze, e la disponibilità di slot liberi.
- L'utente può far richiesta di preservare un posto libero in una delle rastrelliere a sua scelta per un periodo limitato (5m).
- Arrivato alla rastrelliera l'utente si collega con questa.
- Manda la richiesta di deposito
- La rastrelliera comunica al server che l'utente ha richiesto il deposito.
 1. Se l'utente aveva una bici allora il server risponde alla rastrelliera restituendo il tag della bici e procedendo al deposito.
 2. Se non ha una bici il server non avrà tag collegato all'utente e quindi notificherà l'errore e rimanderà un messaggio di errore in app. Nel messaggio di errore si notifica all'utente che non ha una bici e che probabilmente ha sbagliato a fare richiesta (caso in cui l'utente ha cliccato su deposita anziché consegna).
- La rastrelliera apre uno slot e lo comunica all'APP.
- L'utente porta la bici e la posiziona correttamente sullo slot affinché venga letto il tag.
- La rastrelliera manda il tag al server e lo confronta con quello precedente, se sono uguali procede con il deposito altrimenti notifica l'errore e avvisa l'utente.
- La rastrelliera chiude la bici nello slot e manda al server la conferma di terminato deposito, il server risponde confermando e chiudendo le transazioni(billing).
- l'utente riceve in app una notifica di terminata transazione.

5.4.3 Sensori / attuatori

Nel seguito una sezione è dedicata a ciascuno dei sensori e attuatori connessi alla Rastrelliera Intelligente.

5.4.3.1 Rilevatore RFID

5.4.3.1.1.1 Interfaccia fisica

Per quanto riguarda la scelta del lettore RFID la soluzione in prima analisi adottata verte verso una tecnologia di identificazione del TAG non di prossimità (< 4cm), ma di vicinanza (10 cm) in modo tale da non avere vincoli troppo restrittivi tra la distanza fisica tra il rilevatore RFID e il TAG posto sulla bicicletta (vedi *Figura 27*, in cui la distanza massima tra lettore RFID e TAG è di 65 mm).

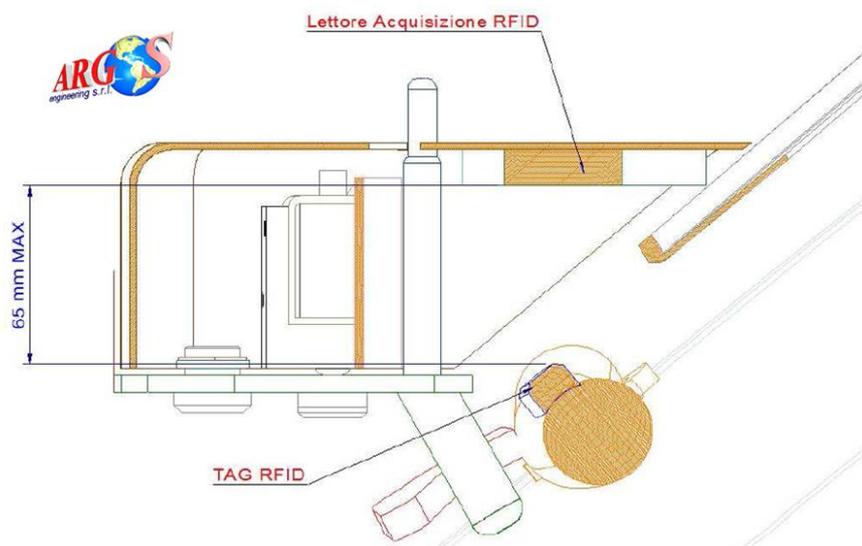


Figura 27 : Sistema locale di identificazione RFID installato sullo stallo della rastrelliera

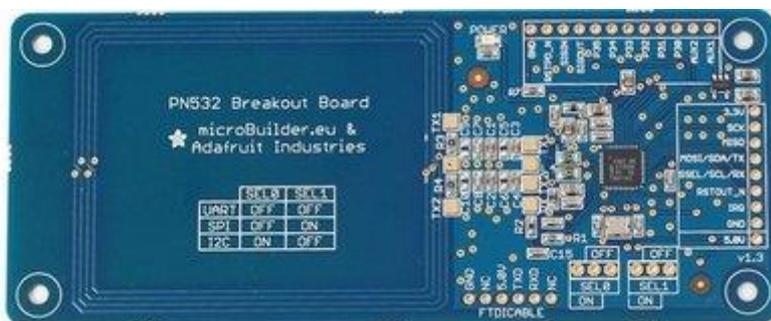


Figura 28 : Sistema locale di identificazione RFID - PN 532 RFID/ NFC (Arduino like)

Il sistema di identificazione RFID scelto è il PN 532 (*Figura 28*) le cui principali caratteristiche sono:

- Protocolli supportati: I2C(default)/SPI/UART;
- Tensione operativa: 3.3 V;
- Compatibile con Arduino Diecimila, Duemilanove, UNO;
- Adatto per letture NFC & TAG MIFARE;
- Frequenza Operativa: 13.56 MHz
- Range di Lettura Massimo: 10 cm
- Basato su modulo PN532
 - ISO/IEC 14443/MIFARE Reader/Writer
 - Felica Reader/Writer

○ ISO/IEC 14443/MIFARE Reader/Writer

Questo chip set è molto potente, e può praticamente fare tutto, come la lettura e la scrittura di tag e carte, comunicare con i telefoni (ad esempio per l'elaborazione dei pagamenti), ed 'agire' come un tag NFC. La frequenza operativa del modulo di lettura è stata scelta come compromesso tra range di lettura, costo e coefficiente di penetrazione del materiale (parametro interessante in quanto il sito di installazione ha materiali metallici nelle vicinanze).

Il protocollo di comunicazione scelto è I^2C che è preferibile per alcune sue caratteristiche:

- segue la logica *master/slave*, ideale in ottica della gestione del traffico tra la motherboard e più sensori RFID;
- permette di raggiungere distanze di qualche metro, a differenza del protocollo SPI;
- in ottica di avere un sistema flessibile alla struttura della rastrelliera, è possibile usare delle estensioni di bus I^2C che agevolano la comunicazione a distanze dell'ordine di decine di metri;
- il numero di pin di interfacciamento è minimizzato, anche lato motherboard: tale aspetto sarebbe stato molto critico facendo uso di comunicazione seriale UART, soprattutto con rastrelliere dotate di un numero di stalli maggiore di 4/5;
- la scheda breakout/shield scelta prevede l'uso di un piedino digitale che fa da notifica di avvenuta identificazione di un TAG RFID;
- il chip PN532 prevede già tutti i traslatori di livello necessari per interfacciare la breakout board con piattaforma Arduino con logica di 5 V.

5.4.3.1.1.2 Interfaccia logica

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

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

La Figura 29 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.

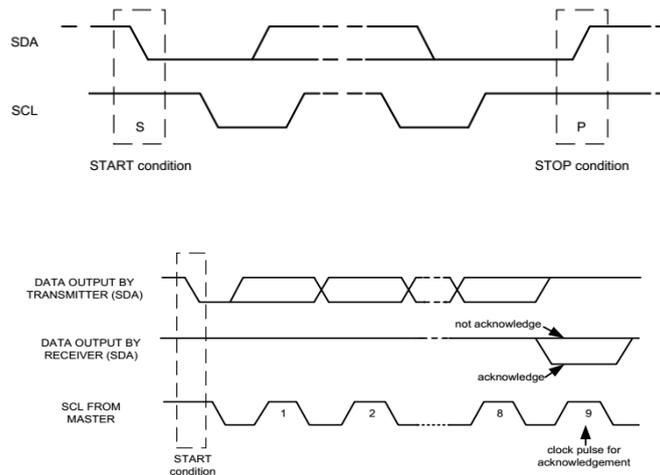


Figura 29 : definizione della fase di START/STOP, ACKNOWLEDGE sul bus I²C

La comunicazione quindi si compone delle seguenti fasi:

- Trasmissione dal *master* di 7 bit di indirizzo dello *slave* + un ottavo bit (R/W) in cui si specifica se il *master* è in lettura o scrittura ;
- Il master attende il messaggio di ACK da slave;
- La comunicazione si interrompe una volta imposte le condizioni di *stop* descritte in [Figura 30](#).

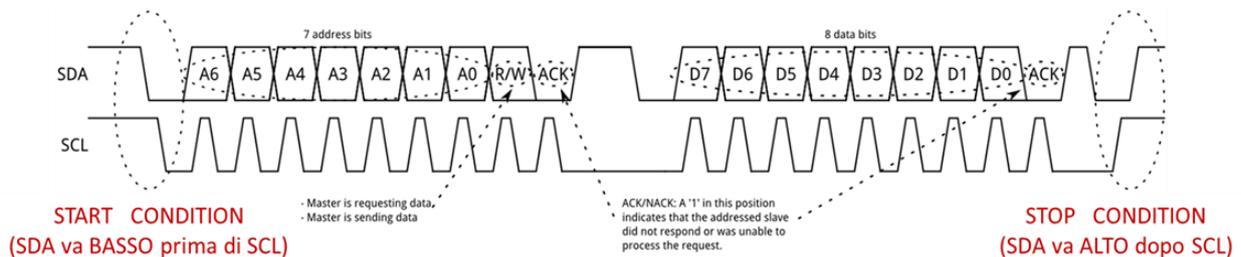


Figura 30 : comunicazione sul bus I²C (indirizzo slave + data)

Il modulo lettore RFID PN532 lavora ad una tensione operativa di 3.3 V ma ci sono dei traslatori di livello interni che permettono la comunicazione I²C con le schede Arduino operative a 5 V (piattaforma).

5.4.3.1.1.3 Architettura

Di seguito ([Figura 31](#)) è riportata l'architettura di principio di funzionamento del kit che equipaggerà la rastrelliera: principalmente esistono due flussi di informazioni di cui uno è relativo al monitoraggio dello stato della rastrelliera in termini di stato del singolo stallo e attuazione del gancio meccanico, mentre il secondo è inerente l'aggiornamento del SSM (Sensor Server and Manager) e la ricezione di notifiche da parte della piattaforma originati dall'interazione tra SSM e utente che è interessato ad accedere alla rastrelliera. Il flusso di informazioni tra piattaforma e SSM segue le regole delle chiamate REST.

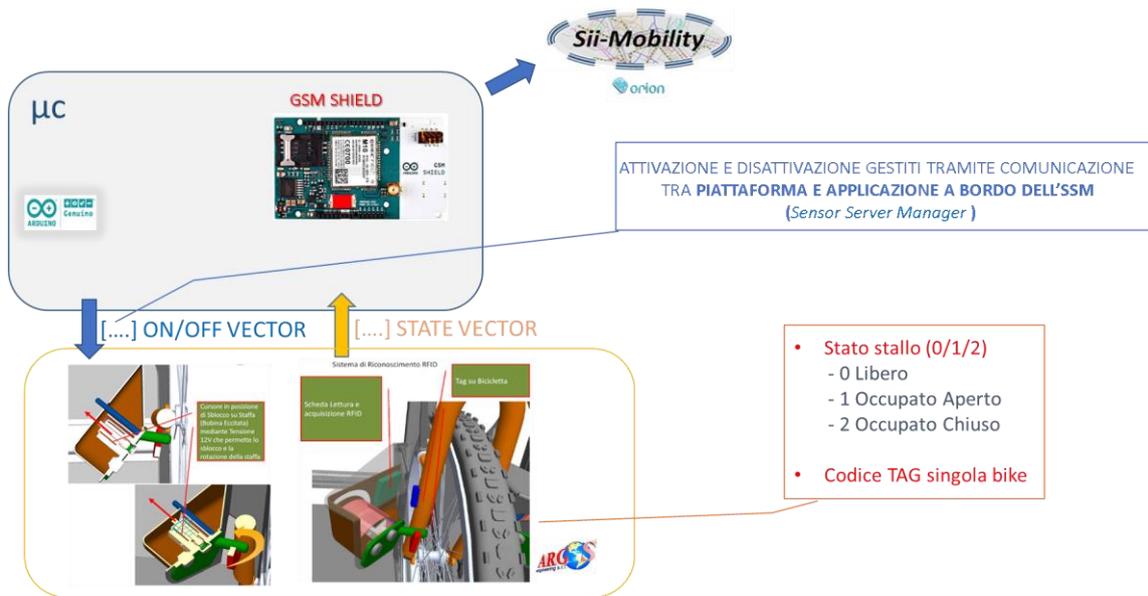


Figura 31 : Architettura generale di principio del sistema elettronico, caso operativo di rastrelliera con singolo stallo

Figura 31 mette in luce anche i servizi che la piattaforma rastrelliera rende disponibili al SSM (vedi anche 5.4.2.4.1) inerenti lo stato dello stallo che può essere:

1. Libero
2. Occupato Aperto
3. Occupato Chiuso

A livello di scambio di informazioni tra piattaforma e sensori, si fa riferimento ai flussi informativi rappresentati in Figura 22, Figura 24, Figura 25.

5.4.3.1.1.4 Specifica di dettaglio

In questo paragrafo di dettaglio si vuole riportare un possibile utilizzo di una estensione del bus dati, da adottare nel caso di rastrelliera di elevata estensione. In particolare, al fine di proporre una soluzione il più possibile flessibile a fronte di una rastrelliera composta da N sensori si può considerare l'applicazione di un estensione del bus dati I²C attraverso l'utilizzo del chip P82B715PN su entrambi i "front end" (Figura 32). La distanza massima di comunicazione su bus I²C è limitata a qualche metro: attraverso l'utilizzo di questo chip è possibile aumentare la distanza fino a decine di metri.

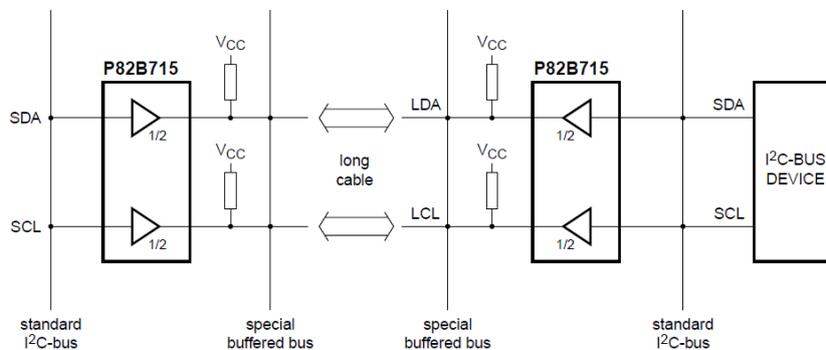


Figura 32 : CHIP P82B715 per estendere il bus I²C

5.4.3.2 Vincolo Meccanico

5.4.3.2.1.1 Interfaccia fisica

Il sistema di Blocco proposto prevede un bloccaggio da un lato della bicicletta e viene attivato facendo ruotare il perno (blu) sull'esterno del supporto: quando la staffa (verde) arriva in posizione all'interno scatta un sistema a molla che blocca la bici sul lato su cui ho eseguito l'operazione (vedi *Figura 33 Sx*).

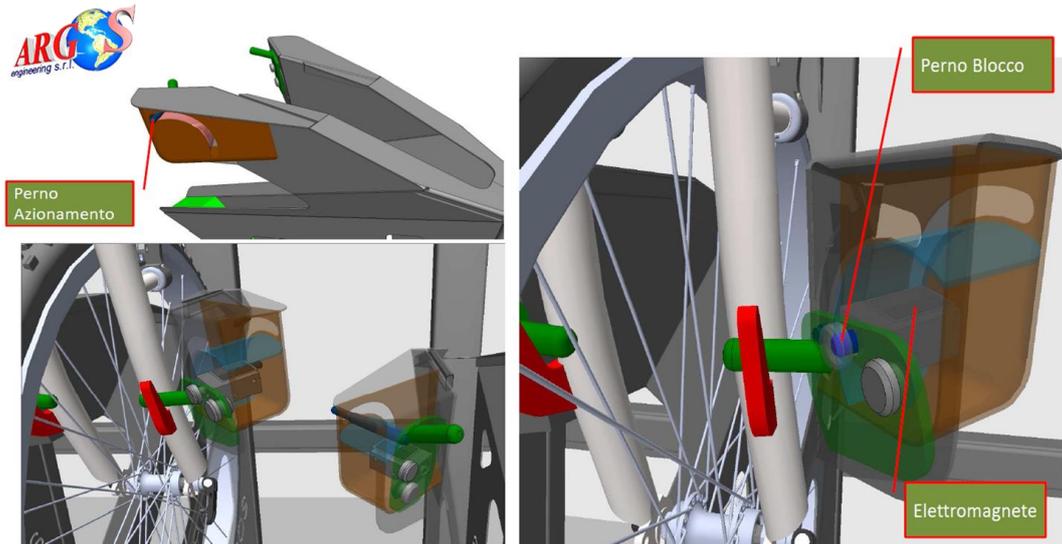


Figura 33 : perno azionamento con intervento esterno (Sx) e elettromagnete dotato di bobina che permette lo sblocco mediante applicazione di una tensione di 12 V (Dx)

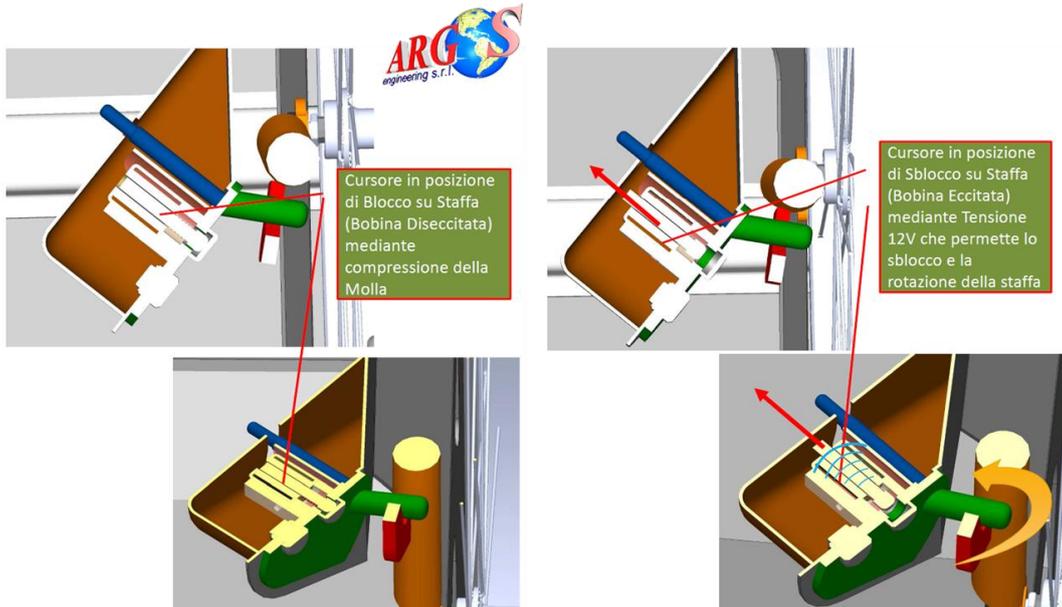


Figura 34 : cursore in posizione di blocco mediante compressione a molla (Sx) e cursore in posizione di sblocco mediante applicazione di eccitazione della bobina tramite una tensione di 12 V

Il perno azionato da una molla è però un cursore di un elettromagnete dotato di bobina elettrica che lavora a trazione e quindi risponde al comando di sblocco soltanto mediante attivazione di una tensione elettrica di fatto generata da una logica di utilizzo del sistema (vedi *Figura 34 Dx*).

L'attuatore tramite solenoide reagisce ad una applicazione della tensione 12 V e, di conseguenza, ai fini del suo controllo, è sufficiente implementare un relè comandato da un pin digitale del μC a bordo del totem (Figura 35).

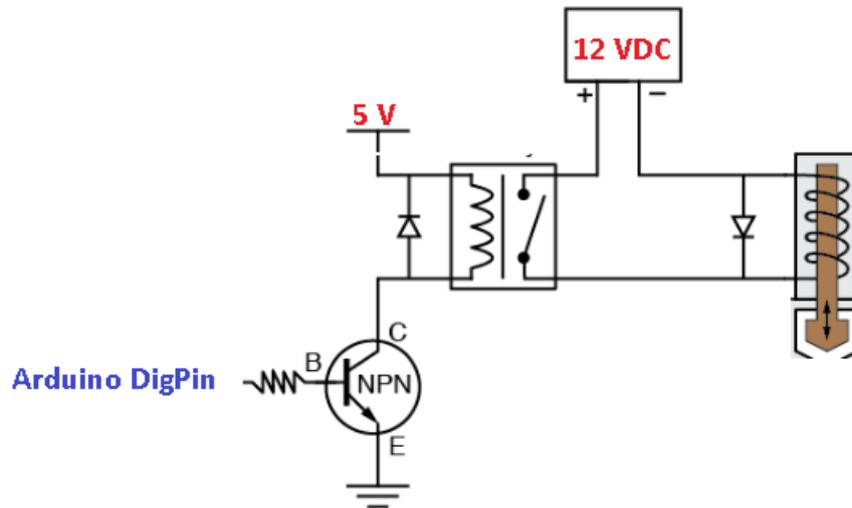


Figura 35 : schema di principio di controllo tramite pin digitale di Arduino del singolo solenoide responsabile del gancio/sgancio della bicicletta allo stallo

Per avere un feedback sull'avvenuto corretto azionamento è necessario dotare ogni stallo di un finecorsa e a tale scopo è stato selezionato un sensore ottico in modo tale da limitare l'impatto con la struttura della rastrelliera ed evitare fenomeni di usura di contatto (tipica di finecorsa a pressione).

Il finecorsa selezionato è il SHARP GP1A57HRJ00F, illustrato in Figura 36 con il suo schema elettronico nel quale sono evidenziati lo stadio di emissione (2 pin) e ricezione (3pin).

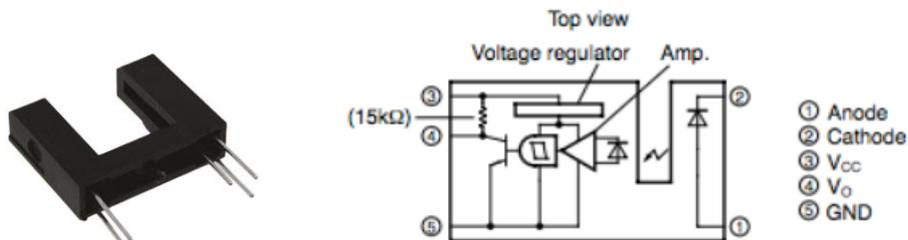


Figura 36 : finecorsa ottico SHARP GP1A57HRJ00F

Per implementare il sensore ottico di finecorsa è stato selezionato la seguente breakout board della Sparkfun in cui i terminali di collegamento sono:

1. Vcc (5V);
2. GND;
3. SIG (Segnale di uscita del finecorsa).

Quando il segnale ottico non è interrotto il pin SIG restituisce un livello di tensione > 4.9 V, mentre in caso contrario inferiore a 0.4 V.



Figura 37 : breakout board per l’implementazione del sensore di finecorsa ottico SHARP GP1A57HRJ00F

Per il dimensionamento dell’attuatore è fondamentale identificare la distanza tra emettitore e ricevitore, che risulta essere pari a 10 mm (vedi Figura 38): il collegamento tra la motherboard e i finecorsa è descritto in Figura 39 , che illustra il collegamento dei 3 pin sopra elencati.

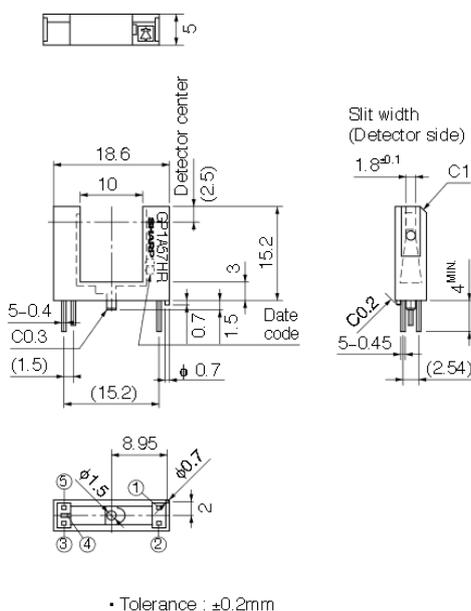


Figura 38 : Datasheet del sensore SHARP GP1A57HRJ00F-distanza tra emettitore e ricevitore

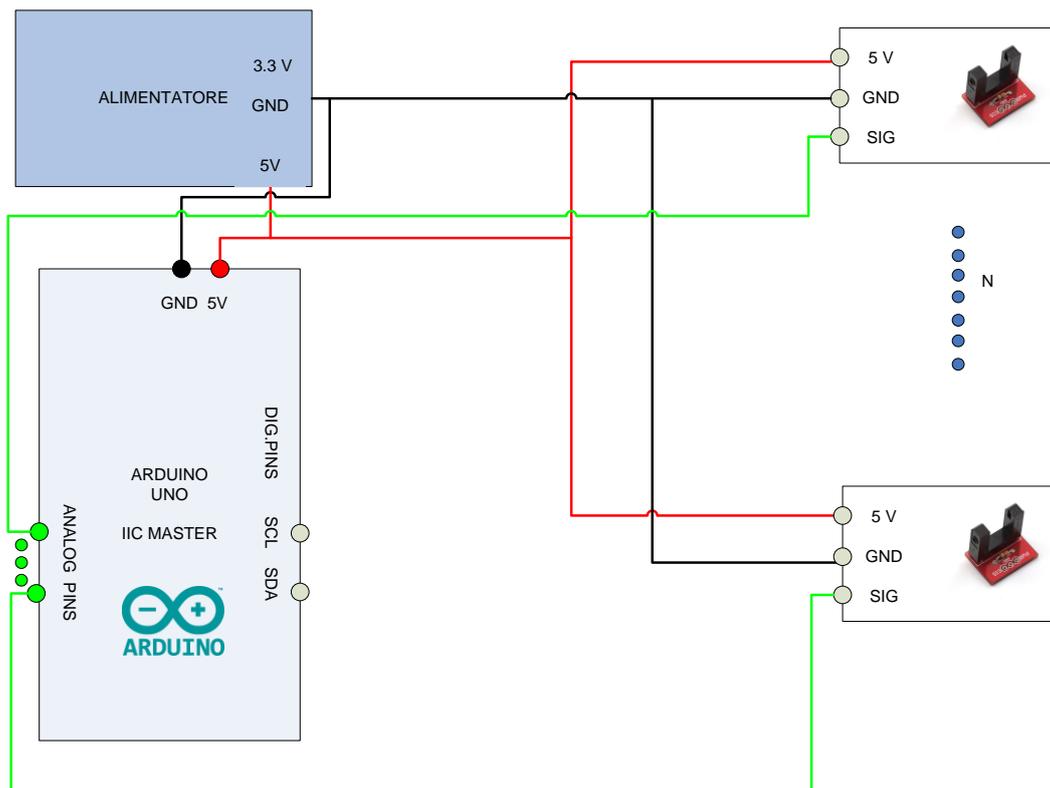


Figura 39: Schema generale della Rastrelliera Intelligente-definizione delle connessioni di lettura del segnale dei finecorsa

5.4.3.2.1.2 Interfaccia logica

Il relè selezionato per pilotare lo stato di eccitazione/diseccitazione del solenoide è compatibile con i livelli logici alto/basso dei pin digitali di Arduino 101 (LOW=0 V, HIGH: 5 V). Inoltre, è stato scelto di utilizzare un relè che è normalmente aperto NO (di conseguenza il gancio è chiuso) che tramite applicazione dei 12 V comporta l’apertura del gancio. Per quanto riguarda i sensori di finecorsa anch’essi sono operativi a 5 V e di conseguenza completamente compatibili con la logica di Arduino UNO.

5.4.3.2.1.3 Architettura

Per quanto riguarda l’architettura fa fede la *Figura 25* alla sezione 5.4.2.3.1.2 e quanto illustrato in *Figura 39*.

5.4.3.2.1.4 Specifica di dettaglio

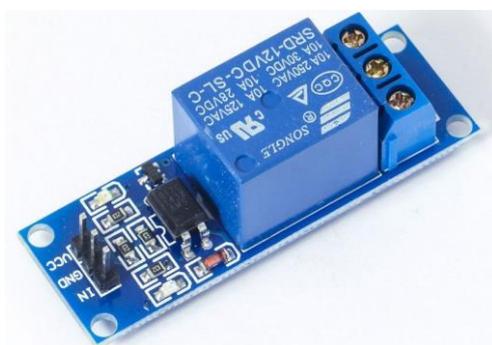


Figura 40 : relè dedito all’azionamento del perno usato per l’aggancio/sgancio della bicicletta alla rastrelliera

Una possibile relè di azionamento del solenoide è illustrato in [Figura 40](#) e le sue principali caratteristiche sono:

- tensione della bobina: 5v dc
- bobina Resistenza: $70 \omega \sim 80 \omega$
- Resistenza di contatto: 100ω max
- tempo di funzionamento: 10msec max
- Tempo di rilascio: 5msec max
- Temperatura ambiente: -25°C a 70°C
- umidità in funzionamento: 45-85% ur
- corrente di funzionamento: 43mA ~ 46ma
- versione corrente: 15mA ~ 18Ma
- pin: 5pin (permette la configurazione NO (*normal open*), NC (*normal close*))
- resistenza di isolamento: ≥ 100 (ohm)
- rigidità dielettrica:
 - tra bobina & contatti: ac 1500V 50hz ~ 60hz / min
 - tra contatti: ac 1000v 50hz ~ 60hz / min
- carico nominale: 10A 250V ac / 10a 125v ac / dc 10a 30v / 10a 28V dc

Per quanto riguarda l'attuatore la scelta deve rispettare due vincoli:

1. Diametro dell'attuatore inferiore ai 10 mm dato che la distanza tra emettitore e ricevitore del finecorsa ottico è di 10mm (vedi [Figura 38](#));
2. Attuatore in configurazione pull in quanto il gancio in condizioni di riposo deve essere chiuso e l'applicazione dei 12 V comporta l'apertura del blocco (vedi [Figura 34](#)).

L'attuatore scelto per l'applicazione è il solenoide tubolare a trazione Saia-Burgess 195224-230, di corsa massima 17.8mm, tensione di alimentazione 12 V c.c., 7W indicato in [Figura 41](#).

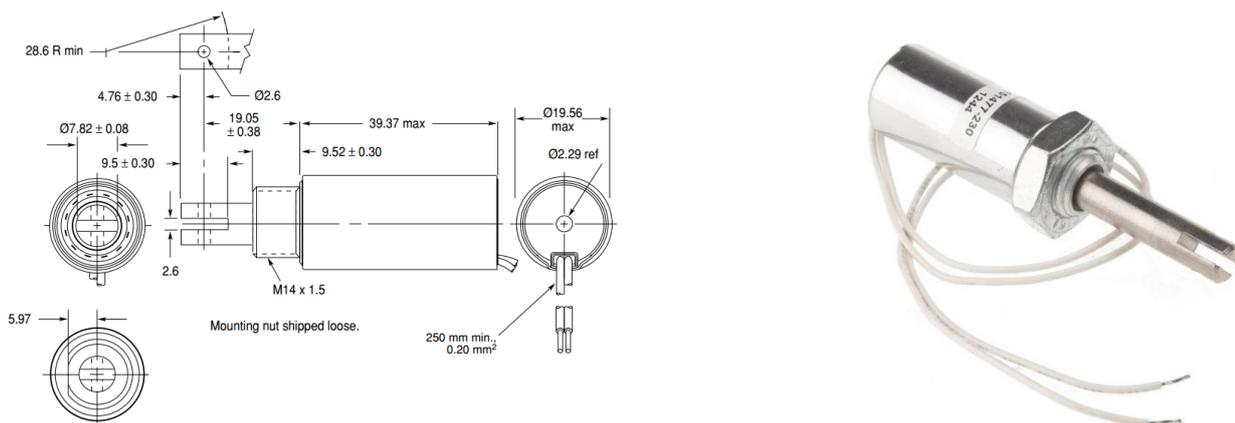


Figura 41: solenoide tubolare a trazione Saia-Burgess (modello 195222-23)