

***Premio nazionale sull'innovazione digitale
ANITEC-ASSINFORM 2022***

**PROGETTO
*SMART TRAFFIC LIGTH FOR ROOBOPOLI
(STLFR)***

INDICE

PREMESSE	3
OBIETTIVI	3
RIFERIMENTI NORMATIVI	4
DESCRIZIONE DEL PROGETTO	4
ORGANIZATIONAL BREAKDOWN STRUCTURE	5
ORGANIZATIONAL BREAKDOWN STRUCTURE	6
RISULTATI ATTESI	7
RISULTATI ATTESI DAL PROGETTO	7
RISULTATI DIDATTICI	8
CARATTERISTICHE DEL SISTEMA DI CONTROLLO	8
ARCHITETTURA HARDWARE	10
CIRCUITI DI INTERFACCIAMENTO	11
RFID READER	11
TAG RFID UHF 868-915 MHz	11
PANNELLO FOTOVOLTAICO 12V 5W	12
AUTOMA ABGS	16
AUTOMA AGAS	17
SCHEMI ELETTRICI	23
SCHEMI A BLOCCHI DEL SISTEMA DI GESTIONE GARAGE VEICOLI	26
ARCHITETTURA DELLA RETE DI TRASMISSIONE DATI	26
SERVER WEB	28
ARCHITETTURA WEB	29
PRESENTATION LAYER	29
BUSINESS LAYER	29
DATA LAYER	29
COSTI	30

PREMESSE

E' noto che da sempre la tecnologia si è interessata allo sviluppo di soluzioni che rendessero più agevole la vita quotidiana ed in particolare più fruibili le infrastrutture, i servizi e più in generale, gli ambienti in ambito urbano così da realizzare delle città a misura d'uomo e possibilmente "intelligenti". La trasformazione della *City* in *Smart* si evidenzia ed articola su cinque aree di intervento:

- *l'economia;*
- *la mobilità;*
- *l'ambiente;*
- *le persone;*
- *lo stile di vita;*
- *la governance.*

Ormai diverse città europee stanno investendo nel modello di "Smart City" che puntano a creare una rete fra cittadini, imprese e amministrazioni pubbliche per fornire migliori servizi e un futuro più sostenibile grazie a un progetto pilota finanziato dal programma "Ue Horizon 2020", il programma quadro dell'UE per la ricerca e l'innovazione.

Secondo gli addetti ai lavori, nell'attuale programmazione europea le opportunità di finanziamento per progetti legati direttamente o indirettamente alle *Smart City* arrivano a circa 456,6 miliardi di euro.

Per promuovere e facilitare l'accesso a questo processo di trasformazione è stata redatta una guida gratuita rivolta alle amministrazioni locali e alle parti interessate: la "**Smart City Guidance Packag**". Il volume è strutturato in sette passaggi che hanno come punto di riferimento i diciassette obiettivi di sviluppo sostenibile 2030 dell'Onu ed è stato realizzato dalla "**European innovation partnership on smart cities and communities**".

Tra le varie sfide prioritarie che l'UE ha identificato nelle quali l'investimento nella ricerca e l'innovazione possono avere un impatto reale a beneficio dei cittadini rientra la *mobilità integrata*.

OBIETTIVI DIDATTICI DEL PROGETTO

Saper lavorare in gruppo è una competenza fondamentale sempre più richiesta nei contesti sia formativi che professionali. La cooperazione, soprattutto in un sistema scolastico che ha sempre privilegiato la competizione e l'individualismo, diventa oggetto di insegnamento attraverso la proposta di esperienze che dimostrino concretamente ai ragazzi che impegnarsi insieme in attività complesse per raggiungere un obiettivo comune è un'esperienza gratificante e vantaggiosa per la crescita e l'apprendimento di ciascuno e, soprattutto, che le abilità sociali sono determinanti per il successo del lavoro del gruppo stesso.

L'obiettivo didattico cardine di questo progetto, oltre quelli di promuovere sintesi creativa, interdisciplinarietà, familiarizzare con le nuove tecnologie e discussione sull'impiego etico delle stesse, consiste proprio nello sviluppare le capacità di lavorare in gruppo. L'insegnante, in questo caso, riveste un ruolo di facilitatore ed organizzatore delle attività, operando in un "ambiente di

lavoro, approfondimento ed apprendimento” in cui gli studenti trasformano le attività del progetto in un processo di *“problem solving di gruppo”*, conseguendo obiettivi la cui realizzazione richiede necessariamente il contributo e l’impegno personale di tutti.

Tutto ciò è stato messo in atto realizzando un percorso di *project based learning*, fondato su un’azienda simulata a cui hanno partecipato ragazzi di classi diverse dell’indirizzo Informatica e Telecomunicazioni articolazione Informatica, che, partendo dall’idea del sistema ideato, procede percorrendo tutte le fasi di sviluppo professionale del progetto consistenti in *analisi delle specifiche, problem solving, individuazione della componentistica elettrica/elettronica, scelta dei sensori/attuatori, sviluppo hardware, sviluppo firmware, integrazione e test logico.*

RIFERIMENTI NORMATIVI

- **D.P.R. N. 495/1992**
Regolamento di esecuzione e di attuazione del nuovo codice della strada. (GU Serie Generale n.303 del 28-12-1992 - Suppl. Ordinario n. 134)
- **DECRETO-LEGGE 10 SETTEMBRE 2021, N. 121 (C.D. DECRETO TRASPORTI).**
Nuove norme riguardanti il Codice della Strada.
- **LEGGE 9 NOVEMBRE 2021, N. 156**
Conversione con modificazioni, del Decreto-Legge 10 settembre 2021, n. 121 (c.d. Decreto Trasporti).
- **D.L. 30 DICEMBRE 2021, N. 228**
Modifiche apportate al Codice della Strada
- **L. 25 FEBBRAIO 2022, N. 15 E DAL D.L. 17 MAGGIO 2022, N. 50**
Conversione del D.L. 30 dicembre 2021, n. 228,

DESCRIZIONE DEL PROGETTO

Il sistema progettato, e in parte realizzato, riguarda il controllo della sequenza dei semafori posti in prossimità di un incrocio in cui oltre, ad implementare il ciclo semaforico standard, è stata sviluppata una sezione che consente di gestire la priorità del transito di automezzi di soccorso e conseguentemente ridurre la probabilità di incidenti dovuti alle situazioni di emergenza.

Il progetto prevede una coppia di transponder per ciascuna corsia che, considerate le ridotte dimensioni del prototipo, nel nostro caso è rappresentata da **RFID HF** che lavorano a **13,56 MHz** e consentono di rilevare i corrispondenti **TAG** passivi ad una distanza dell’ordine dei 10 cm.

Nel caso di *smart city* reali si possono utilizzare degli **RFID READER LONG RANGE UHF** che lavorano a **840 MHz** e che con specifici **TAG passivi** di tipo *on metal* è possibile raggiungere distanze di comunicazione addirittura fino a 30/35 metri.

Un primo transponder è installato sulla corsia di ciascun semaforo predisposto a rilevare il segnale di avvicinamento di un mezzo di soccorso, equipaggiato con il **TAG**, il secondo subito a valle dell’incrocio sulla medesima corsia per rilevare la fine dell’impegno dell’incrocio da parte del mezzo di soccorso

Al momento il sistema, considerati i tempi brevissimi a disposizione per la progettazione, è stato sviluppato supponendo di avere un incrocio a quattro vie solo con possibilità di prosecuzione sulla stessa corsia (non può svoltare né a destra né a sinistra)

Si sta già pensando ad una nuova release che gestisce le diverse direzioni di prosecuzione del mezzo di soccorso.

E' stata sviluppata anche una sezione di gestione di malfunzionamenti dovuti essenzialmente a guasti degli **RFID** che consente di escludere la gestione delle emergenze e far funzionare solo la sezione del ciclo semaforico di base.

Il sistema progettato prevede quattro tipologie di interventi:

- *modifica parziale del firmware del **RooboKart***
- *upgrading hardware del **RooboKart***
- *Progettazione e sviluppo ex novo del firmware di gestione della sequenza semaforica*
- *upgrading hardware del circuito di controllo del ciclo semaforico*

ORGANIZATIONAL BREAKDOWN STRUCTURE

In questa sezione si formalizzano le linee di riporto tra i componenti del *project team* che nasce da una attenta analisi delle attività del progetto.

La **OBS** è particolarmente utile all'interno del progetto in quanto chiarisce i livelli di coordinamento, i punti di controllo organizzativi e l'ambito di autonomia dei ragazzi coinvolti.

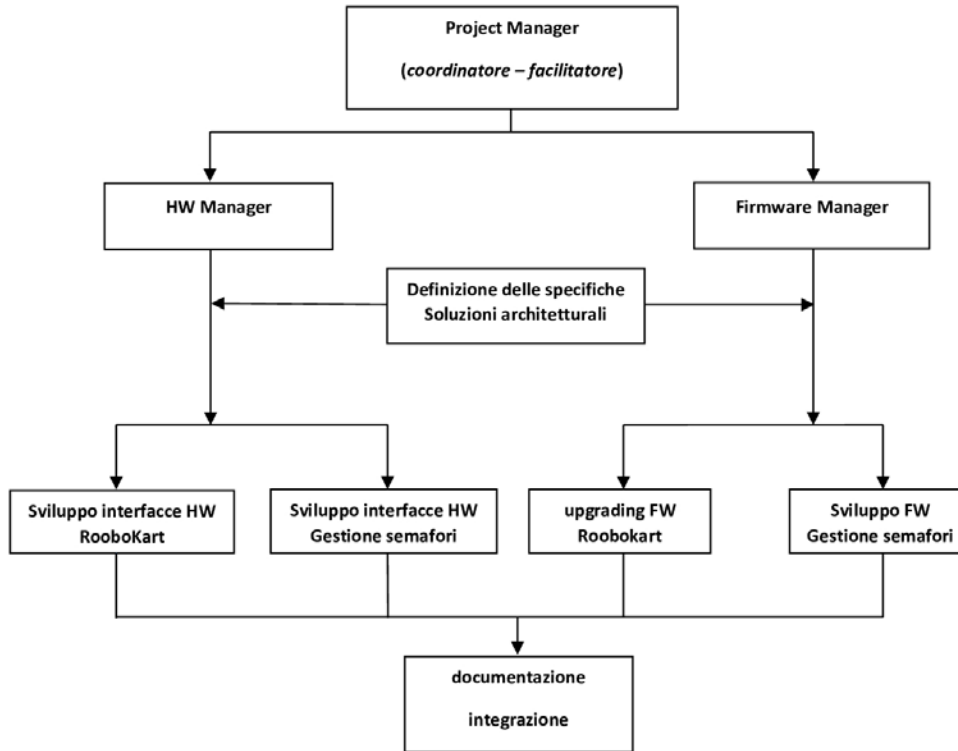
Il processo utilizzato per creare la **OBS** può essere così riassunto:

- *individuare le attività necessarie allo svolgimento del lavoro.*
- *individuare le competenze necessarie allo svolgimento di ciascuna attività.*
- *valutare quali e quante risorse coinvolgere nel progetto.*
- *Posizionare le risorse nell'organigramma di progetto in modo da facilitarne l'operatività ed il processo decisionale ai fini dei risultati del progetto.*

La creazione della **OBS** è quindi importante per:

- *ufficializzare le risorse coinvolte nel progetto;*
- *per chiarire le modalità di riporto tra i ragazzi coinvolti;*
- *per comprendere il livello di integrazione tra i gruppi;*
- *per facilitare il ruolo del Project Manager per quanto riguarda le attività di monitoraggio e coordinamento.*

Fig. 1: ORGANIZATIONAL BREAKDOWN STRUCTURE



PIANIFICAZIONE DELLE ATTIVITA' DEL PROGETTO

Di seguito vengono riportate le varie fasi di sviluppo del progetto corredato da un diagramma di Gantt che ne definisce le tempistiche

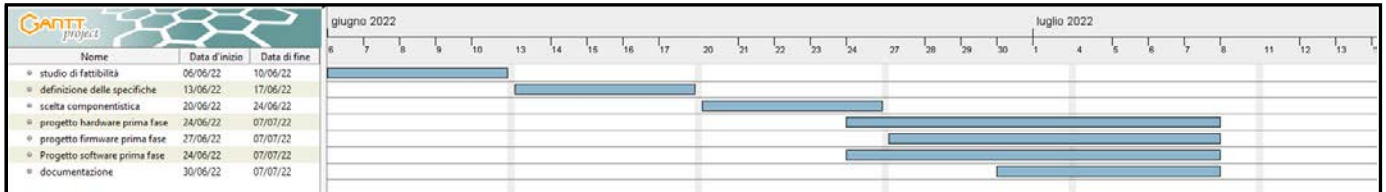
- *analisi fattibilità;*
- *definizione delle specifiche;*
- *Individuazione della componentistica da utilizzare;*
- *Progettazione e Sviluppo Firmware di gestione della sequenza semaforica standard e in situazione di emergenza;*
- *Progettazione e Sviluppo modulo Firmware di upgrading **RooboKart**;*
- *Progettazione e Sviluppo Hardware per gestione semafori;*
- *Integrazione;*
- *Test logico funzionale;*

Una volta definite le fasi del progetto, sono stati calendarizzati due *briefing* (intesi come momento di pianificazione, approfondimento e condivisione) giornalieri di circa due ore in cui il *Team* si riunisce per fare il punto della situazione, confrontarsi su eventuali problematiche sorte durante lo sviluppo del progetto e definire eventuali aggiustamenti in corso d'opera.

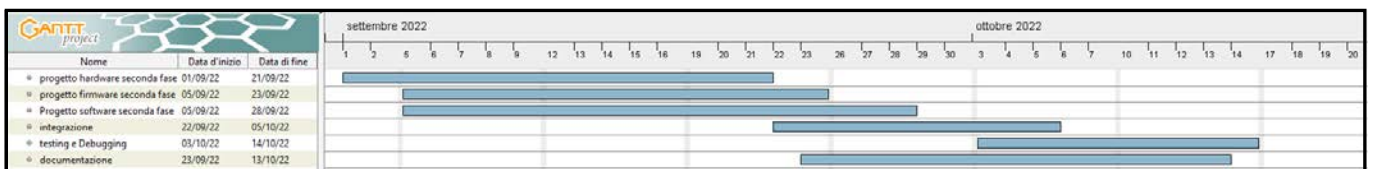
Fig. 2: PIANIFICAZIONE DELLE ATTIVITA' DEL PROGETTO

Il Diagramma di Gantt del progetto è uno strumento utile in quanto consente di visualizzare e tracciare le tempistiche e l'avanzamento delle attività.

- Prima fase:



- Seconda fase:



RISULTATI ATTESI

Risultati attesi dal progetto

Indichiamo, nel seguito, i principali risultati attesi per ogni fase. Tali risultati sono volutamente molto specifici in modo da poter essere verificati, e costituiranno, quindi, lo strumento primario per la valutazione dell'avvenuto raggiungimento degli obiettivi del progetto e controllarne l'avanzamento.

- *definizione delle specifiche:*
 - *partendo dall'idea si definisce ciò che deve fare il sistema;*
- *Individuazione della componentistica da utilizzare:*
 - *Partendo dalle specifiche si individueranno i componenti hardware e le loro caratteristiche tecniche;*
- *Progettazione e Sviluppo Firmware di gestione della sequenza semaforica standard e in situazione di emergenza:*
 - *Realizzazione del firmware per l'acquisizione dei dati provenienti dai **RFID READER** e conseguente attuazione dei colori corrispondenti ai quattro semafori;*
- *Progettazione e sviluppo dei moduli Firmware del **RooboKart** da inserire nel software già esistente:*
 - *Realizzare il firmware di gestione della modalità **emergenza** del **RooboKart** ponendo particolare attenzione ad utilizzare tecniche di programmazione che consentano di non incidere sul software già esistente;*
- *Progettazione e Sviluppo Hardware per gestione semafori:*

- Realizzazione degli schemi elettrici dei circuiti di interfaccia necessarie per il collegamento degli RFID READER alla scheda ST.
- Progettazione dell'APP:
 - Realizzazione dell'APP per ANDROID che consente lo switch di modalità di funzionamento del **RooboKart**
- Integrazione;

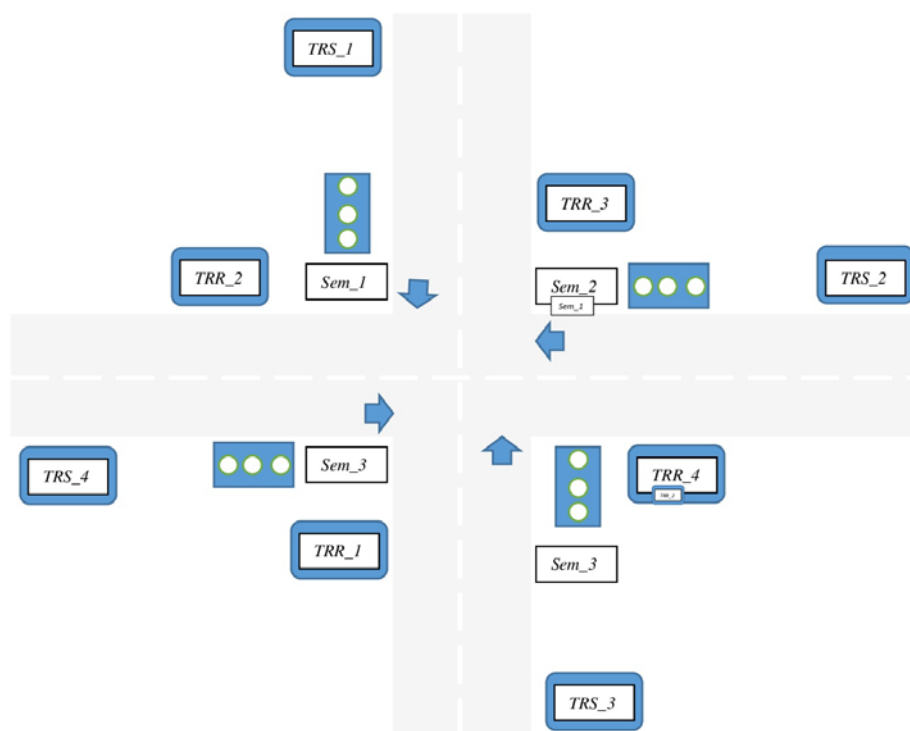
Il successo del progetto potrà essere misurato in termini del raggiungimento dei risultati attesi elencati.

Risultati didattici

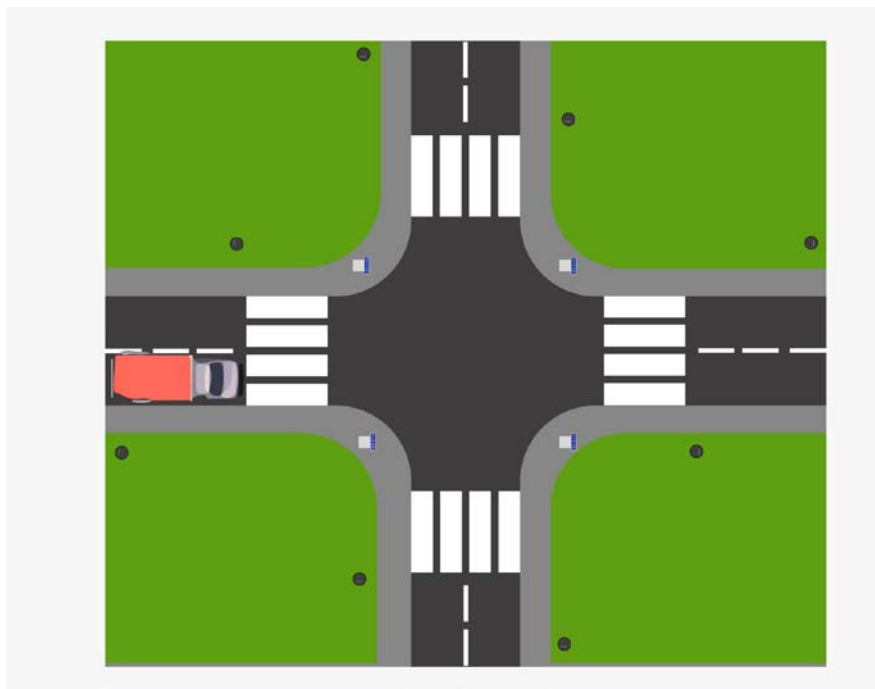
- comprendere le funzioni che svolgono i componenti scelti per la realizzazione del sistema;
- conoscere le caratteristiche dei sensori scelti per l'acquisizione dei dati;
- legami disciplinari, concettuali e operativi, tra Meccanica, Fisica, Informatica, Elettronica e Gestione Progetti;
- saper organizzare i dati di un problema da risolvere mediante schemi o grafici e tradurre gli algoritmi con linguaggi di programmazione;
- capacità di collaborazione e di lavoro in gruppo.

CARATTERISTICHE DEL SISTEMA DI CONTROLLO

FIG. 3: SCHEMA LOGICO FUNZIONALE DEL SISTEMA



- *TRsX* -> *Trsponder x set allarme*
- *TRRx* -> *Trsponder x reset allarme*



ARCHITETTURA HARDWARE

- modifica hardware e firmware del **RoobKart** :
 - Il sistema prevede un **TAG RFID HF** a bordo del mezzo di soccorso (nel nostro caso un **RooboKart**) in cui viene memorizzato il proprio numero identificativo.
 - Il **RooboKart** prevede a bordo un modulo **Bluetooth HC05**, letto ciclicamente da un **Thread**, che consente di settarlo dinamicamente, tramite uno *smartphone*, in *modalità mezzo di emergenza*.
 - Per la gestione di un lampeggiatore a led è previsto un **Ticker**
 - Per la gestione di una sirena da implementare con il buzzer (sirena bitonale 950 HZ per 0.6 s e poi 700 Hz pr 0.4 s) è previsto un **Ticker**.
- Upgrading *hardware* del circuito di controllo del ciclo semaforico
 - A bordo carreggiata, in prossimità degli incroci semaforizzati (nel caso reale circa 100 m), è prevista l'istallazione di un **RFID Reader HF NFC PN532** funzionante a **13.56 MHZ**.
 - Nel caso di transito di un mezzo di soccorso equipaggiato con **TAG** la sua rilevazione, da parte del **RFID READER**, verrà captata dalla scheda **NUCLEO F 401 RE**, preposta al controllo del ciclo semaforico, che attiverà un automa di gestione delle emergenze con conseguente variazione del ciclo semaforico di base.

SCHEDA CONTROLLO DELLA SEQUENZA SEMFORICA

Il cuore del sistema di acquisizione e condizionamento di segnali è costituito da una scheda **STMicroelectronic NUCLEO F 401RE**.

caratteristiche

- *STM32F401RET6 in LQFP64 package*
- *ARM®32-bit Cortex®-M4 CPU with FPU*
- *84 MHz max CPU frequency*
- *VDD from 1.7 V to 3.6 V*
- *512 KB Flash*
- *96 KB SRAM*
- *GPIO (50) with external interrupt capability*
- *12-bit ADC with 16 channels*
- *RTC*
- *Advanced-control Timer*
- *General Purpose Timers (7)*
- *Watchdog Timers (2)*
- *USART/UART (4)*
- *I2C (3)*
- *SPI (3)*
- *SDIO*
- *USB 2.0 OTG FS*

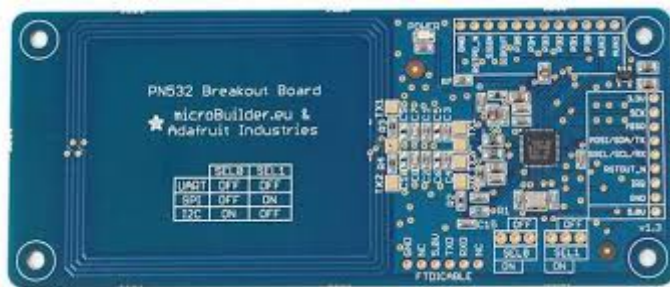
CIRCUITI DI INTERFACCIAMENTO

RFID READER HF 13.56 MHz

PN532 è un modulo di trasmissione **RFID HF (Radio Frequency Identification)** che opera su **13,56 MHz** con funzionalità di microcontrollore basata su un core **80C51**.

Per effettuare lo scambio di informazioni con i sistemi host sono implementate diverse interfacce:

- *Interfaccia SPI;*
- *Interfaccia I2C;*
- *UART seriale (simile a RS232 con livelli di tensione 0 e PVDD);*



Una volta alimentata ed effettuato il setup, la scheda leggerà i **TAG** che rientrano nel suo campo di azione (circa 10 cm). Consente anche la scrittura dei **TAG**.

caratteristiche

- *Microprocessore 80C51 con ROM da 40 kbyte e 1 kbyte RAM*
- *Driver di uscita bufferizzati per collegare un'antenna con un numero minimo di componenti esterni*
- *Supporta ISO 14443A / MIFARE*
- *Supporta ISO 14443B solo in modalità lettura/scrittura*
- *Possibilità di comunicare sull'interfaccia RF superiore a 424 kbaud utilizzando circuiti analogici esterni*
- *Interfaccia SPI*
- *Interfaccia I2C*
- *UART seriale ad alta velocità*
- *Timer programmabile*
- *Alimentazione da 2,7 a 5,4 V*
- *Porte IO specifiche per il controllo di dispositivi esterni*

TAG RFID HF 13.56 MHz

I **TAG HF** sono trasponder con all'interno chip ed antenna **RFID HF** detti anche di classe "**High Frequencies media**" è normalizzata dagli standard di riferimento **EPC**; esistono sia di tipo passivo che attivo; è una classe valida in tutto il mondo anche se la potenza massima e le bande di frequenza incontrano limiti definiti dalle autorità dei singoli paesi; i **TAG** di questa classe possiedono una distanza di lettura maggiore e viene principalmente utilizzata per trasponder attivi nel controllo di accessi, veicoli, etc.

Ogni **TAG** viene fornito con un **TID (Truly Unique ID)** che non può essere modificato.



Riportiamo di seguito le caratteristiche del TAG individuati per il progetto.

SPARKFUN 74HC4951 8 CHANNEL MUX/DEMUX BREAKOUT



Spark Multiplexer Breakout fornisce l'accesso a tutti i pin e le caratteristiche del **74HC4951**, un multiplexer/demultiplexer analogico a 8 canali. Il **74HC4951** consente di trasformare quattro pin I/O in otto segnali multifunzione, selezionabili individualmente, che possono essere utilizzati per eseguire qualsiasi operazione, dal pilotaggio di otto LED al monitoraggio di otto potenziometri. Il **74HC4951** può funzionare come *multiplexer* o *demultiplexer* e dispone di otto canali di ingressi/uscite selezionabili. L'instradamento del segnale comune verso l'I/O viene impostato controllando digitalmente tre linee di selezione.

PANNELLO FOTOVOLTAICO 12V 5W



In silicio policristallino con il 18% di efficienza della cella, nessun diodo; tuttavia, questi possono essere installati in aggiunta. Superficie in vetro solare temperato ESG con rivestimento resistente alle intemperie. Certificato TÜV. Strutture in alluminio anodizzato. Prestazioni fino a 10 anni con il 90% di rendimento e fino a 25 anni con l'80%.

Voltaggio	12 volt
Amperaggio	0.28 A
Potenza massima	5 watt
Certificazione	SGS TÜV IEC61215, IEC61730

ARCHITETTURA SOFTWARE

Linguaggio di sviluppo firmware

Il firmware del sistema **STLFR** è stato sviluppato utilizzando *linguaggio C++*. Come ambiente di sviluppo si è utilizzato **MBED** di **ARM**, una piattaforma che ben si presta allo sviluppo su *cloud* di applicazioni **IoT** basate su microcontrollori **ARM** e che dispone di svariate librerie di oggetti standard collaudate e che si evolve ed aggiornano continuamente.

Il firmware della parte di upgrading del software esistente del **RooboKart** è stato sviluppato utilizzando una *toolchain* in **ECLIPSE**.

Semafori

Il software per la gestione del ciclo semaforico intelligente prevede un automa principale, indicato con **ABGS** (*Automa Base Gestione Semafori*), che gestisce il ciclo semaforico

standard e un secondo automa, indicato con **AGAS** (*Automa Gestione Allarmi Semaforici*), che viene chiamato in caso di transito di un mezzo di soccorso su una delle quattro corsie che comporta una variazione del ciclo semaforico in modo da attivare il colore verde sulla corsia su cui sta transitando e portare a rosso quelli sulle altre corsie.

Nel caso di attivazione di un allarme l'automata **ABGS** attiva l'automata **AGAS** e si pone in uno stato di *idle* attendendo la restituzione del controllo da parte dell'automata **AGAS** che in base ad un *flag*, settato a seguito dell'attivazione di uno dei trasponder **TRRx** ed interpretato dallo stato **Gall**, deciderà se tornare passare allo stato **SI** o **S3**.

Per gestire l'eventuale contemporaneità di allarmi si utilizza una **FIFO** circolare.

L'automata di gestione degli allarmi è indicizzato in modo da poter funzionare in caso di attivazione da parte di qualsiasi dei quattro **TRsx** e di provenienza da qualsiasi dei quattro stati dell'automata **ABGS**.

Quindi in caso di allarme, il controllo dei semafori viene passato all'automata **AGAS** fornendogli due parametri:

- *Lo stato dell'automata di base in cui si è verificato l'allarme*
- *la coda dei trasponder attivati*

L'automata **AGAS** continuerà a girare fin quando non si svuoterà la coda degli allarmi.

In questo caso il controllo verrà restituito all'automata **ABGS** nello stato consequenziale alla cessione dell'allarme.

L'automata **AGAS** per gestire correttamente la sequenza di allarme dei semafori accede ad una matrice di indicizzazione utilizzando i parametri **TRsx** (trasponder che ha generato la transizione) e **Stato_x** (stato da cui ha origine la transizione) individuando l'indice della sequenza semaforica di allarme corrispondente che verrà attivata e mantenuta fin quando non verrà attivato un trasponder **TRRx** (trasponder di reset) dal **TAG** che ha generato l'allarme.

Le informazioni relative al **TRsx** e allo **Stato_x** suddetti sono passate con una variabile *uint8_t* gestita bit a bit. La struttura del dato è:

I° nibble - (Bit₇ - Bit₆ - Bit₅ - Bit₄): TRRx

- *x001 -> TRR1*
- *x010 -> TRR2*
- *x011 -> TRR3*
- *x100 -> TRR4*

II° nibble - (Bit₃ - Bit₂ - Bit₁ - Bit₀): Stato_x

- *0001 -> Stato_0*
- *0010 -> Stato_1*
- *0011 -> Stato_2*
- *0100 -> Stato_3*

Bit₇ - Bit₆ - Bit₅ - Bit₄ - Bit₃ - Bit₂ - Bit₁

Nel caso di *overflow* della coda degli allarmi verrà disabilitato l'automata **AGAS** ed acceso un led lampeggiante di segnalazione di guasto.

L'automa **ABGS** continuerà a gestire la sequenza standard senza più attivare l'automa **AGAS**.

La gestione della coda viene effettuata utilizzando un vettore di *uint8_t* di dimensione 8.

Gestione FIFO allarmi

L'idea di partenza è di creare una area di memoria degli allarmi, e delle funzioni per scrivere e leggere dati in questo buffer (*uint8_t*).

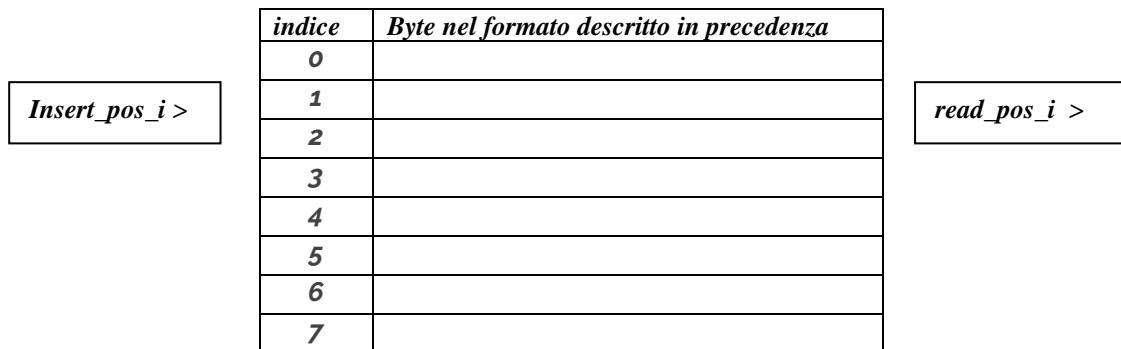
Verranno utilizzati due indici:

- *Insert_pos_i->*: indice del primo elemento vuoto
- *read_pos_i ->* : indice primo elemento inserito

e una variabile che tiene conto degli elementi inseriti:

- *cont_allarm*

di seguito è riportato un modello d'esempio:



Inizialmente:

- *Insert_pos_i-> = 0*
- *read_pos_i -> = 0*
- *cont_allarm = 0*

quando viene inserito un allarme incrementiamo *insert_pos_i* e *cont_allarm*

quando viene prelevato un allarme incrementiamo *read_pos_i* e decrementiamo *cont_allarm*

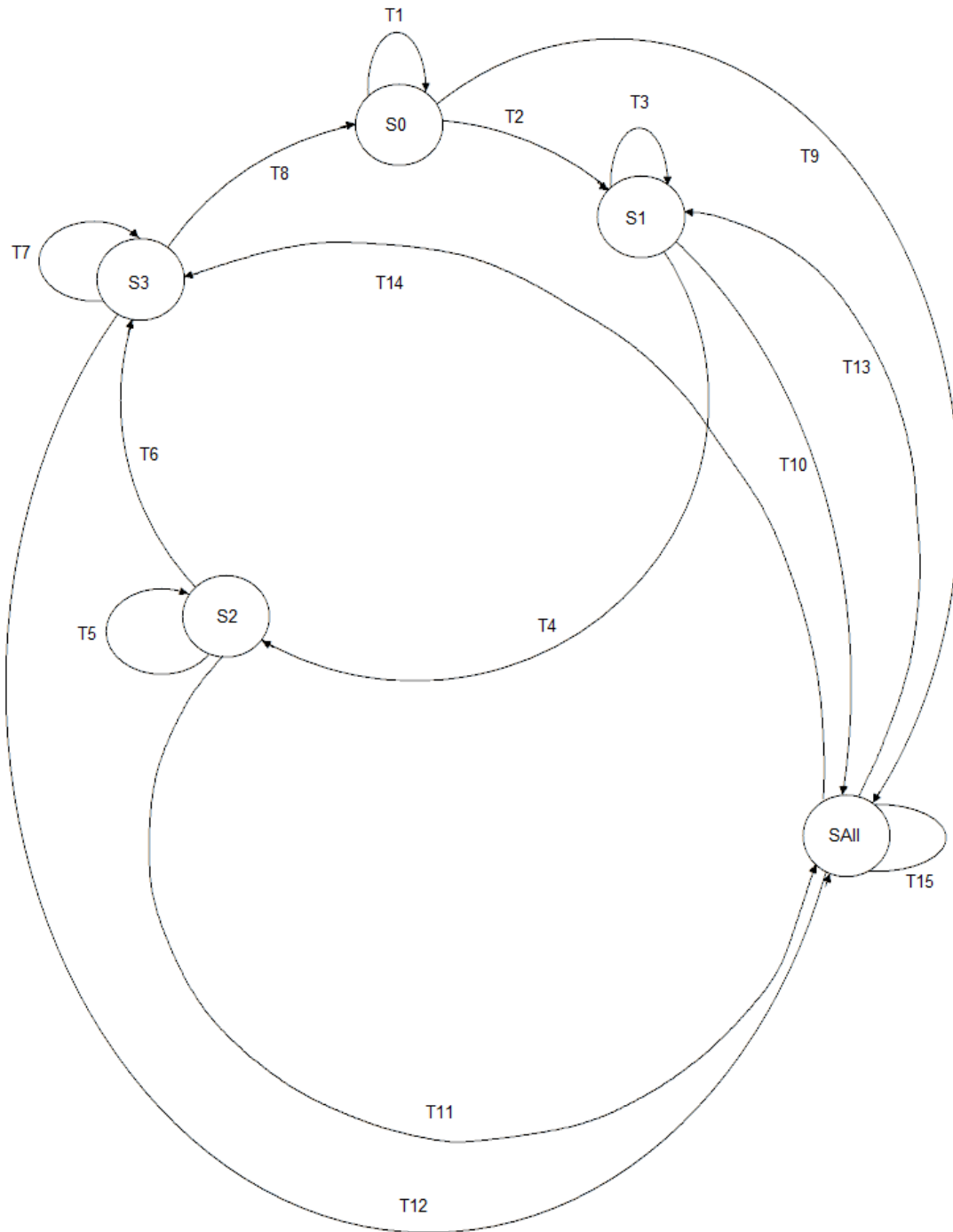
Se dovesse verificarsi che *cont_allarm = 7* e *read_pos_i -> = 0* ed arriva un altro allarme avremo un *buffer overflow* con conseguente disabilitazione dell'automa **AGAS**.

<i>trsponder</i>	<i>Codice hex I* nibble</i>
TRS1	<i>0x01</i>
TRS2	<i>0x02</i>

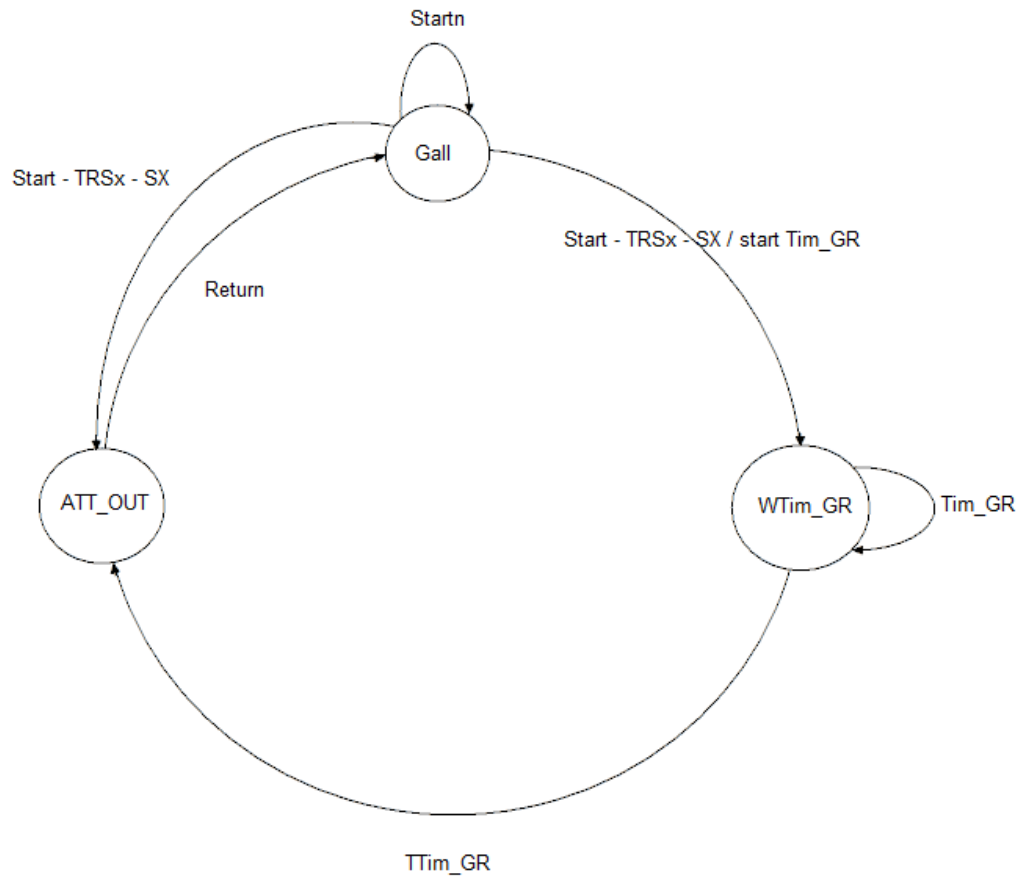
<i>TRS3</i>	<i>0x03</i>
<i>TRS4</i>	<i>0x04</i>

<i>Stato</i>	<i>Codice hex II° nibbel</i>
<i>S0</i>	<i>0x00</i>
<i>S1</i>	<i>0x01</i>
<i>S2</i>	<i>0x02</i>
<i>S3</i>	<i>0x03</i>

AUTOMA ABGS



AUTOMA AGAS



Automa ABGS

Tabella transizione degli stati

<i>INP / Sx</i>	$\overline{Tim_V_R}$	<i>Tim_V_R</i>	$\overline{Tim_G}$	<i>Tim_G</i>	$\overline{TRS1}$	<i>TRS1</i>	$\overline{TRS2}$	<i>TRS2</i>	$\overline{TRS3}$	<i>TRS3</i>	$\overline{TRS4}$	<i>TRS4</i>	<i>ABGS_stato_ret_all = 1</i>	<i>ABGS_stato_ret_all = 3</i>
<i>S0</i>	<i>S0</i>	<i>S1</i>	-	-	<i>S0</i>	<i>SAll</i>	<i>S0</i>	<i>SAll</i>	<i>S0</i>	<i>SAll</i>	<i>S0</i>	<i>SAll</i>	-	-
<i>S1</i>	-	-	<i>S1</i>	<i>S2</i>	<i>S1</i>	<i>SAll</i>	<i>S1</i>	<i>SAll</i>	<i>S1</i>	<i>SAll</i>	<i>S1</i>	<i>SAll</i>	-	-
<i>S2</i>	<i>S2</i>	<i>S3</i>	-	-	<i>S2</i>	<i>SAll</i>	<i>S2</i>	<i>SAll</i>	<i>S2</i>	<i>SAll</i>	<i>S2</i>	<i>SAll</i>	-	-
<i>S3</i>	-	-	<i>S3</i>	<i>S1</i>	<i>S3</i>	<i>SAll</i>	<i>S3</i>	<i>SAll</i>	<i>S3</i>	<i>SAll</i>	<i>S3</i>	<i>SAll</i>	-	-
<i>SAll</i>	-	-	-	-	-	-	-	-	-	-	-	-	<i>S1</i>	<i>S3</i>

Tabella trasformazione delle uscite

<i>INP</i> <i>/</i> <i>Sx</i>	$\overline{Tim_V_R}$	<i>Tim_V_R</i>	$\overline{Tim_G}$	<i>Tim_G</i>	$\overline{TRS1}$	<i>TRS1</i>	$\overline{TRS2}$	<i>TRS2</i>	$\overline{TRS3}$	<i>TRS3</i>	$\overline{TRS4}$	<i>TRS4</i>	<i>ABGS_stato_ret_all = 1</i>	<i>ABGS_stato_ret_all = 3</i>
<i>S0</i>	SEM1 Vgr SEM3 Vgr SEM2 vgr SEM4 vgr	SEM1 vGr SEM3 vGr SEM2 vgr SEM4 vgr	-	-	-	-	-	-	-	-	-	-	-	-
<i>S1</i>	-	-	SEM1 vGr SEM3 vGr SEM2 vgr SEM4 vgr	SEM1 vgr SEM3 vgr SEM2 vGr SEM4 vGr	-	-	-	-	-	-	-	-	-	-
<i>S2</i>	SEM1 vgr SEM3 vgr SEM2 Vgr SEM4 Vgr			-	-	-	-	-	-	-	-	-	-	-
<i>S3</i>	-	-	SEM1 vgr SEM3 vgr SEM2 vGr SEM4 vGr	SEM1 Vgr SEM3 Vgr SEM2 vgr SEM4 vgr	-	-	-	-	-	-	-	-	-	-
<i>SAll</i>	-	-	-	-	-	-	-	-	-	-	-	-	SEM1 vGr SEM3 vGr SEM2 vgr SEM4 vgr	SEM1 vgr SEM3 vgr SEM2 vGr SEM4 vGr

Automa AGAS

Tabella transizione degli stati

(è strutturata diversamente dalla precedente per motivi di dimensione)

Input				Stato		
start	Tim GR	TRSx	Sx	GAll	WTim_GR	ATT_OUT
0	x	x	x	GAll	-	-
1	x	0001	0000	ATT_OUT	-	-
1	x	0001	0001	ATT_OUT	-	-
1	x	0001	0010	WTim_GR	-	-
1	x	0001	0011	WTim_GR	-	-
1	x	0010	0000	WTim_GR	-	-
1	x	0010	0001	WTim_GR	-	-
1	x	0010	0010	ATT_OUT	-	-
1	x	0010	0011	ATT_OUT	-	-
1	x	0011	0000	ATT_OUT	-	-
1	x	0011	0001	ATT_OUT	-	-
1	x	0011	0010	WTim_GR	-	-
1	x	0011	0011	WTim_GR	-	-
1	x	0100	0000	WTim_GR	-	-
1	x	0100	0001	WTim_GR	-	-
1	x	0100	0010	ATT_OUT	-	-
1	x	0100	0011	ATT_OUT	-	-
X	Tim_GR_n	-	-	-	WTim_GR	-
x	Tim_GR_s	-	-	-	ATT_OUT	-
x	x	x	x	-	-	GAll

Tabella trasformazione delle uscite

(è strutturata diversamente dalla precedente per motivi di dimensione)

Input				Stato	
start	Tim GR	TRSx	Sx	SWTim_GR	SATTOUT
0	x	x	x	-	-
1	x	0001	0000	-	null
1	x	0001	0001	-	SEM1 VA GS RS SEM3 VA GS RS
1	x	0001	0010	SEM2 VS GA RS SEM4 VS GA RS	-
1	x	0001	0011	SEM2 VS GA RS SEM4 VS GA RS	-
1	x	0010	0000	SEM1 VS GA RS SEM3 VS GA RS	-
1	x	0010	0001	SEM1 VS GA RS SEM3 VS GA RS	-
1	x	0010	0010	-	null
1	x	0010	0011	-	SEM2 VA GS RS SEM4 VA GS RS
1	x	0011	0000	-	null
1	x	0011	0001	-	SEM1 VA GS RS SEM3 VA GS RS
1	x	0011	0010	SEM2 VS GA RS SEM4 VS GA RS	-
1	x	0011	0011	SEM2 VS GA RS SEM4 VS GA RS	-

1	x	0100	0000	SEM1 VS GA RS SEM3 VS GA RS	-
1	x	0100	0001	SEM1 VS GA RS SEM3 VS GA RS	-
1	x	0100	0010	-	null
1	x	0100	0011	-	SEM2 VA GS RS SEM4 VA GS RS
x	Tim_GR_s	-	0001	-	SEM2 VA GS RS SEM4 VA GS RS
x	Tim_GR_s	-	0011	-	SEM1 VS GA RS SEM3 VS GA RS

Nello stato **GAll** vengono valutati:

- lo stato di provenienza **Sx**
- il trasponder che ha generato la transizione **TRSx**

per determinare le attuazioni da mettere in atto sui diversi semafori e il nuovo stato in cui deve portarsi l'automata **ABGS**.

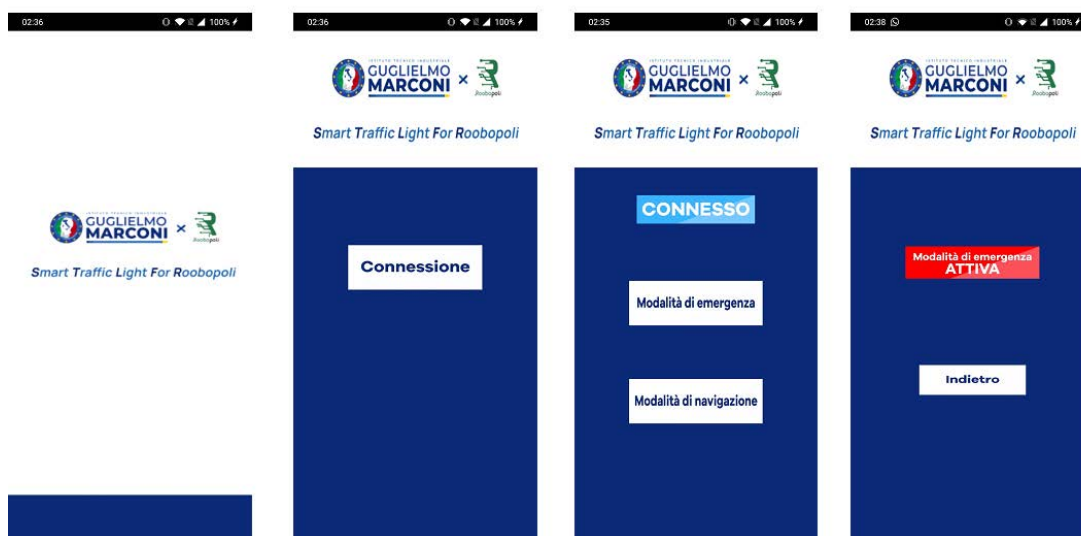
Modulo firmware **RooboKart**

Il modulo relativo allo *switch* di modalità del **RooboKart**, dovendo per ovvi motivi impattare il meno possibile con il firmware già sviluppato, si basa su tre blocchi funzionali:

- **Ticker** per controllo *blink* lampeggiante
- **Ticker** per controllo sirena bitonale (*buzer*)
- **Thread** per la comunicazione *bluetooth*

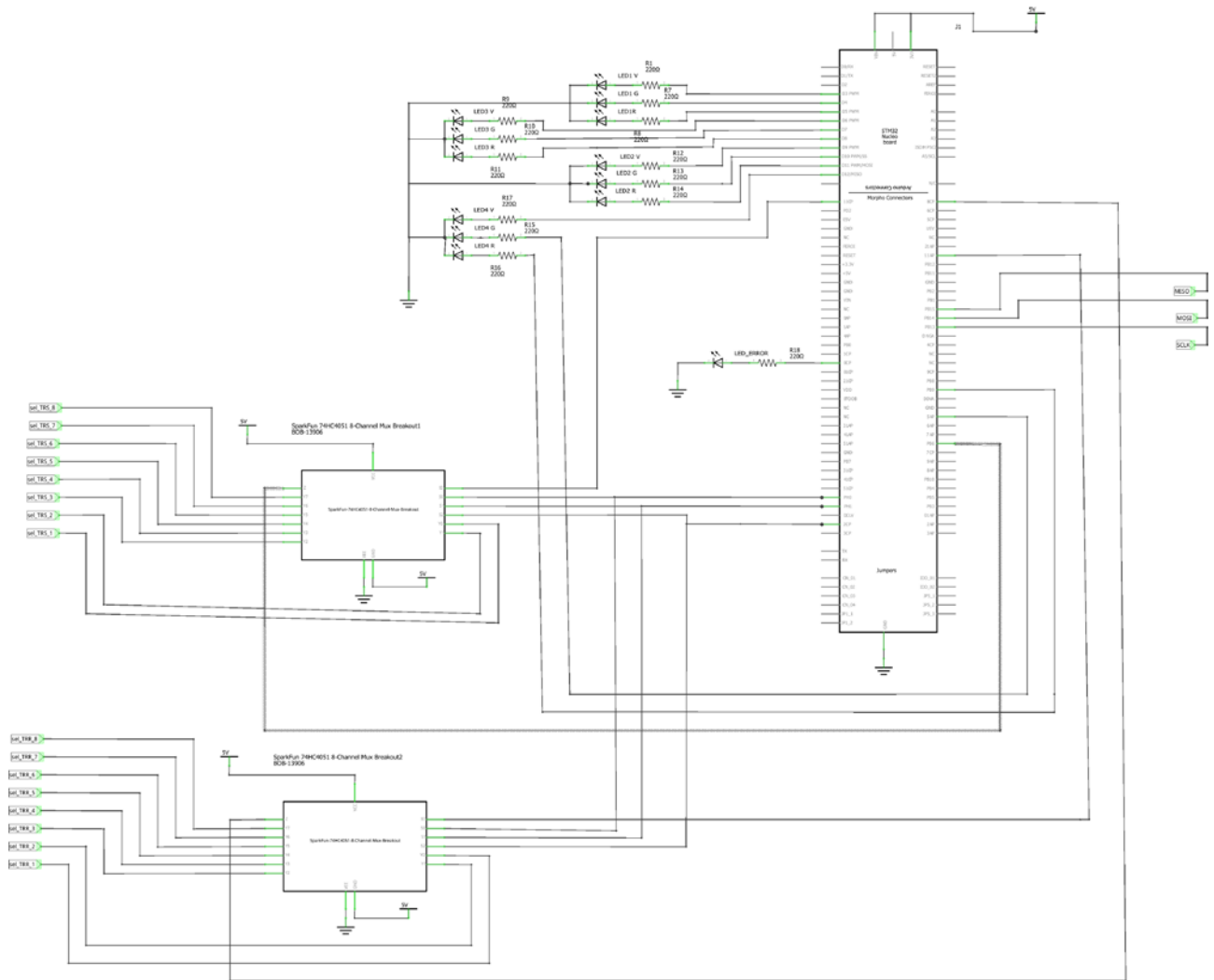
Modulo sviluppo APP ANDROID

E' stata sviluppata un'**APP** per **ANDROID** per effettuare lo *switch* di modalità tramite *bluetooth*



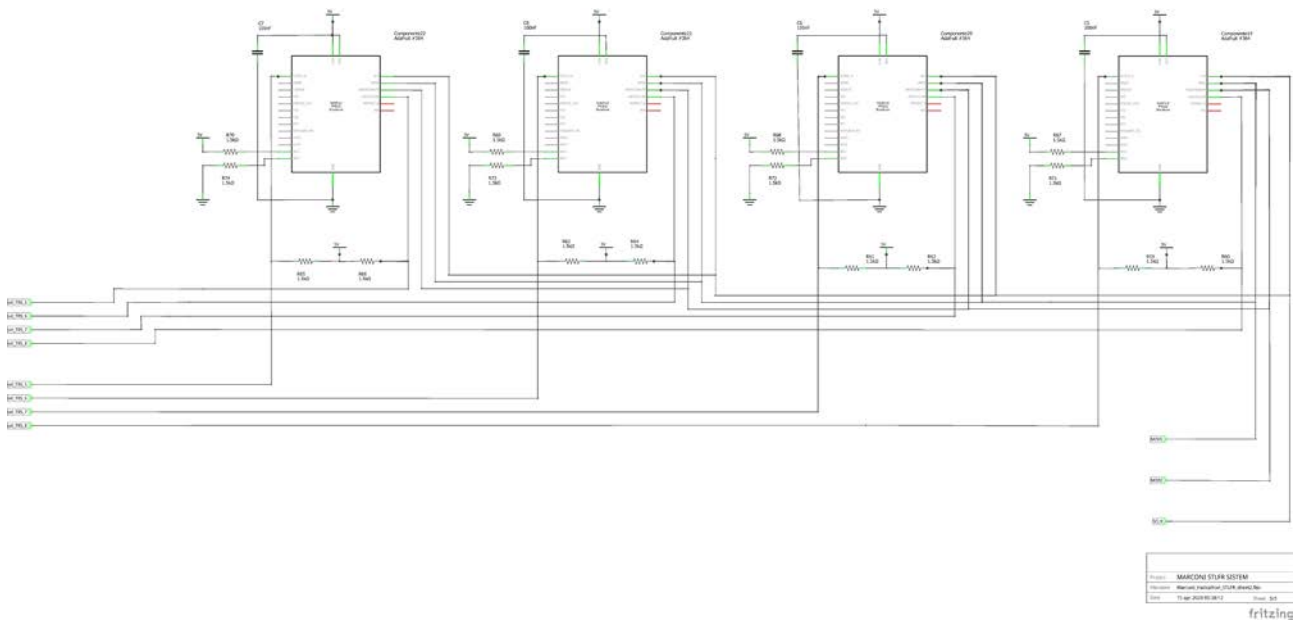
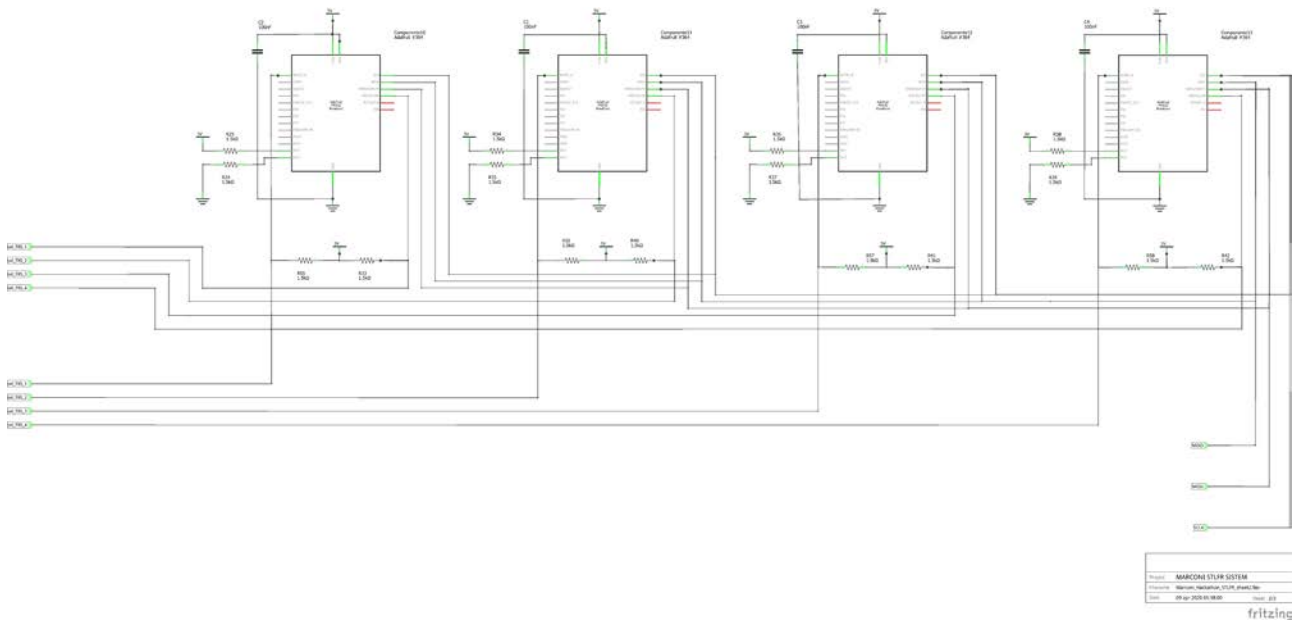


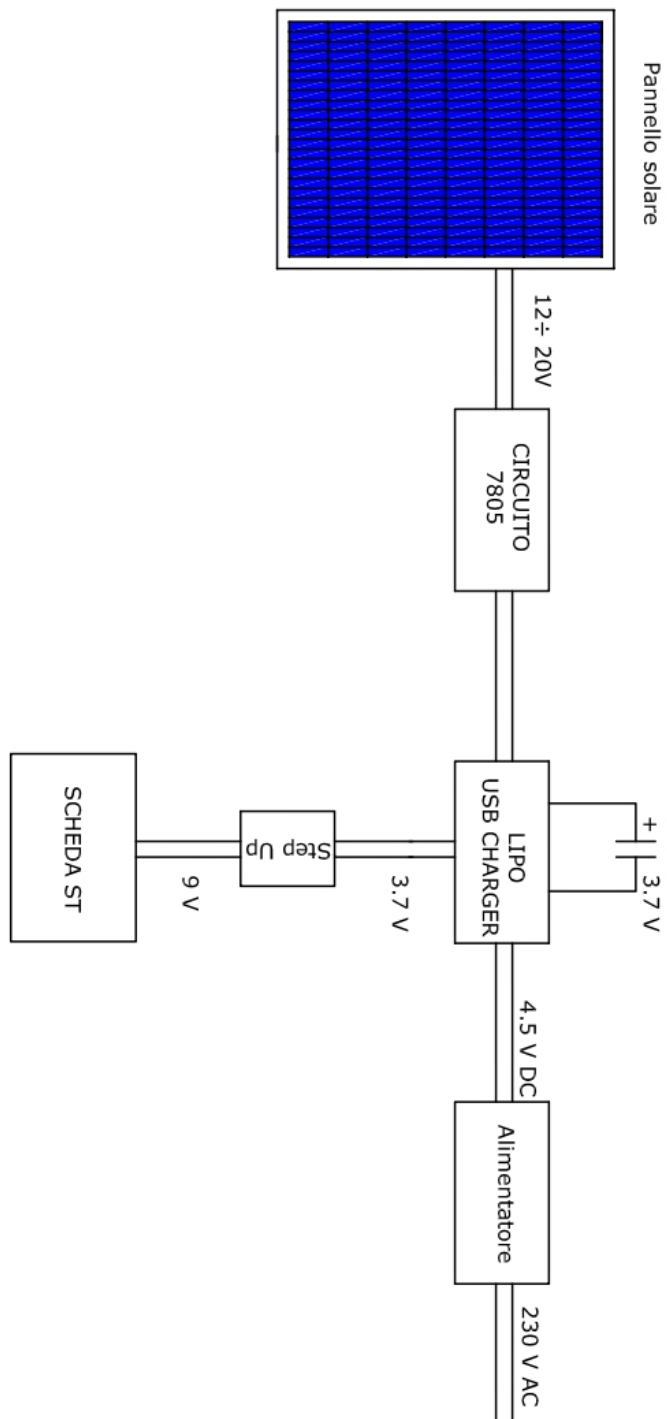
SCHEMI ELETTRICI



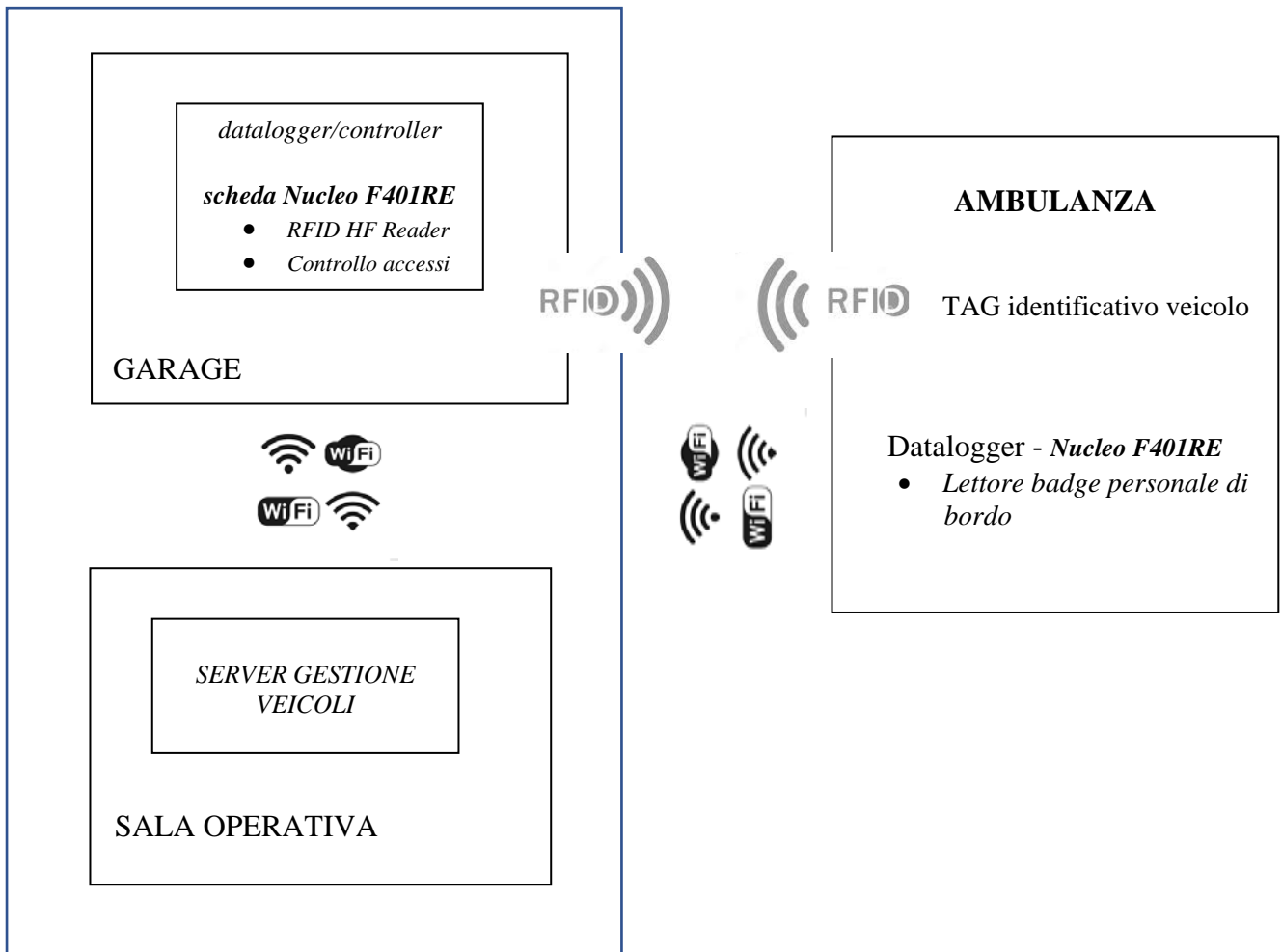
Project	MARCONI STLFR SYSTEM
Filename	Marconi_Hackathon_STLFR_sheet1.kicad
Date	07 apr 2020 10:39:01 sheet: 1/2

fritzing





SCHEMA A BLOCCHI DEL SISTEMA DI GESTIONE DEL GARAGE VEICOLI



ARCHITETTURA DELLA RETE DI TRASMISSIONE DATI

L'architettura utilizzata come già detto in precedenza sfrutta l'infrastruttura di rete della scuola ed è pertanto del tipo client server basata su socket TCP.

Il TCP permette la trasmissione delle informazioni in entrambe le direzioni. I sistemi informatici che comunicano tramite TCP possono pertanto inviare e ricevere dati contemporaneamente, proprio come avviene durante una conversazione telefonica. Le unità di trasmissione fondamentali utilizzate dal protocollo sono i segmenti (pacchetti) che, oltre al carico utile (ossia il messaggio effettivo), possono contenere anche informazioni di controllo e sono limitati a una dimensione di 1.500 byte. L'instaurazione e l'interruzione delle connessioni, classificabili come connessioni punto-a-punto,

nonché la trasmissione stessa dei dati vengono acquisite dal software TCP nella pila di protocolli di rete del rispettivo sistema operativo.

Il software TCP viene controllato dalle varie applicazioni rete, come browser web o server, attraverso interfacce specifiche, laddove ciascuna connessione deve essere sempre identificata da due punti terminali chiaramente definiti (client e server). A questo riguardo, non ha alcuna rilevanza quale lato assuma il ruolo di client e di server – l'importante è che il software TCP abbia a disposizione per ciascun punto terminale una coppia ordinata e univoca formata da indirizzo IP e porta.

Quindi è caratterizzata da due software distinti (che lavorano su due diversi host) uno lato serve (PC) e l'altro lato client (schede data logger/controller). Il server si mette in attesa di una richiesta da servire, il client effettua tale richiesta. I client comunicano con l'unico server, mentre il server comunica con tutti i client distribuiti sui diversi piani.

Nel seguito faremo riferimento al termine IP nel senso di IPv4.

L'handshake a tre vie: come funziona in dettaglio l'instaurazione della connessione TCP

I presupposti per instaurare una connessione TCP valida sono i seguenti: entrambi i punti terminali devono disporre già di un indirizzo IP univoco (IPv4) e avere dichiarato e abilitato la porta desiderata per la trasmissione dei dati. Mentre il primo funge da caratteristica identificativa, la seconda è rilevante per l'assegnazione delle connessioni alle applicazioni client e server concrete da parte del sistema operativo.

Poiché l'instaurazione della connessione tramite il Transmission Control Protocol prevede in totale tre passaggi, per questo processo è entrata in uso la denominazione “**handshake a tre vie**”.

Ancora prima che vengano trasferiti i primi dati, generalmente il mittente e il destinatario si accordano riguardo alla **dimensione massima dei segmenti TCP da inviare** (Maximum Segment Size – MSS). Di norma sono possibili fino a 1500 byte per segmento; di questi, almeno 20 byte devono essere previsti per l'header TCP e altri 20 byte per l'header IP, in modo tale che rimangano **1.460 byte**. Se si desidera una dimensione personalizzata, questa deve essere specificata (come indicato sopra) tramite il campo delle opzioni, eseguendo le dovute sottrazioni per la parte del carico utile.

Le caratteristiche principali del TCP possono essere riassunte come segue:

- Il TCP è orientato alla connessione e permette una comunicazione alternata tra due punti terminali in base al cosiddetto handshake a tre vie.
- Il TCP è affidabile, in quanto assicura che tutti i dati vengono trasmessi in modo completo e possano essere ricomposti dal destinatario nell'ordine corretto.
- Il TCP prevede l'invio dei dati in singoli segmenti che possono avere una dimensione massima di 1.500 bytes (compreso header).
- Nel modello OSI, il TCP viene classificato al livello di trasporto (layer 4).
- Nella maggior parte dei casi, il TCP si basa sul protocollo Internet (IP), pertanto spesso si parla anche di pila di protocolli TCP/IP.
- L'header TCP ha una dimensione standard di 20 byte – a cui possono aggiungersi fino a 40 byte di opzioni aggiuntive.

SERVER WEB

Il progetto prevede lo sviluppo di un sito web responsive (HTML5 e CSS3) in modo da gestire l'identificativo dell'ambulanza, la tipologia di allestimento, i nominativi del team a bordo della stessa, data e ora di partenza e di ritorno.

Il database è formato da sette tabelle di cui due di tramite.

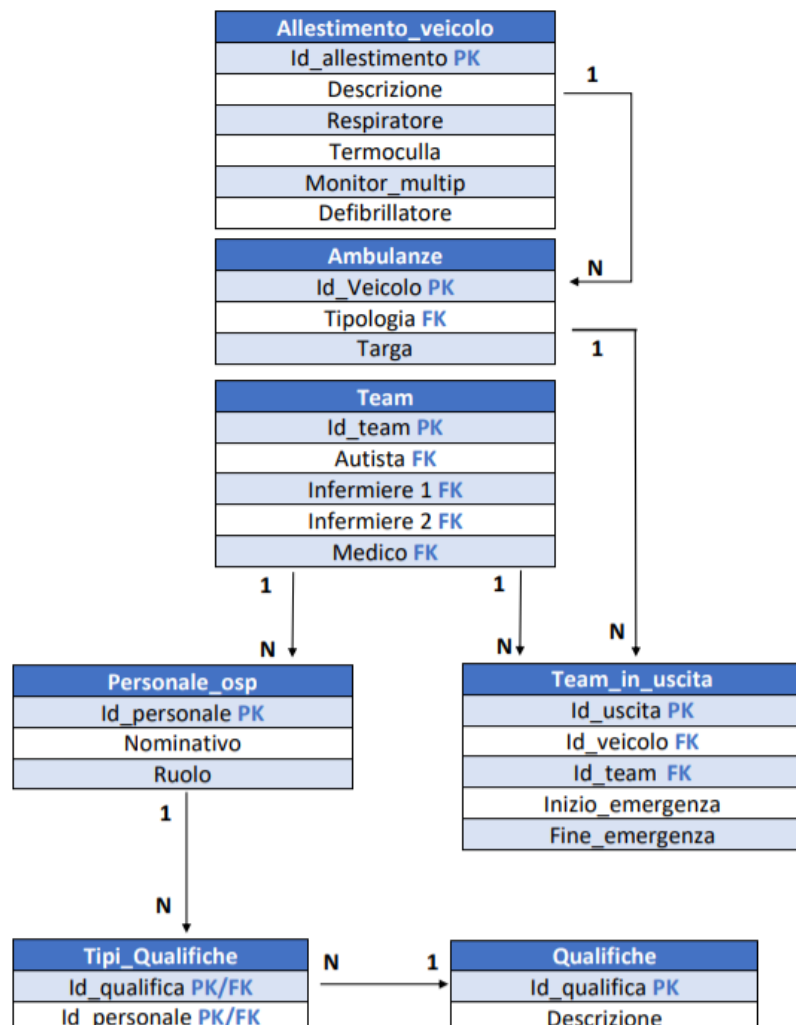
Le tabelle principali sono tre:

- Allestimento_veicolo (contenente l'identificazione dell'attrezzatura presente a bordo delle ambulanze)
- Ambulanze (contenente i dati delle varie ambulanze)
- Personale_osp (contenente i dati del personale di bordo)

Modello E/R:

Il modello E/R progettato, utilizzando un set definito di simboli, consente di gestire le interconnessioni tra Entità, Relazioni ed attributi

- *Allestimento veicolo di soccorso*
- *Ambulanze*
- *Personale di bordo*



ARCHITETTURA WEB

Per lo sviluppo web, si è utilizzata un'architettura a tre livelli :

- Livello Presentazione (detto front-end)
- Livello Logico (detto Business Logic)
- Livello Dati (detto Back-end)

PRESENTATION LAYER:

Il front-end del progetto è rappresentato da un Sito Web scritto principalmente in linguaggio HTML e che presenta alcune funzionalità di CSS.

Questo, essendo un Sito Web che deve essere accessibile solamente dai terminali dell'ospedale, si troverà su un server dedicato con accesso limitato.

La pagina web ha il compito di mettere a schermo varie informazioni quali:

- Lista del personale
- Lista dei veicoli
- Pagine per l'aggiunta di personale nuovo o di nuovi veicoli.

BUSINESS LAYER:

Il Business Layer del progetto è rappresentato da un codice scritto in PHP che comunica con il livello di presentazione (e quindi con l'HTML), comunica con il livello Database (e quindi con MySQL), e fa da tramite per alcune richieste quali:

- Connessione al database;
- Select per la visualizzazione delle liste sia delle Ambulanze che del Personale; - Comandi che
- permettono l'aggiunta di nuovo personale o di nuovi veicoli.

DATA LAYER:

Il Data Layer del progetto è rappresentato da un database scritto in codice SQL, questo contiene tutti i dati di cui si ha bisogno sia per gestire le richieste fatte dal Business Layer.

Il tutto è stato creato in modo da permettere una facile manipolazione dei dati.

COSTI

Il costo stimato per la realizzazione del prototipo per il controllo di un singolo incrocio è di circa €400.

Le ore uomo impiegate nello sviluppo del progetto sono state stimate in circa 80 distribuite.

Il costo stimato per la produzione in serie per 1000 pezzi è il seguente:

Tab. 1: COSTI PER 1000 UNITA'

Centro di costo	quantità	Prezzo €
masterizzazione PCB doppia faccia	1000	1.700,00
montaggio schede	1000	400,00
test e collaudi schede elettroniche	1000	550,00
Materiale	1000	400.000,00
	Totale	402.650,00