# **ESP32 GPS-TRACKER**

*Giovanni Garofalo*

*A13002365*

[*giovanni.garofalo2@studenti.unicampania.it*](mailto:giovanni.garofalo2@studenti.unicampania.it)

*Saverio Valentino*

## Sommario

[**ESP32 GPS-TRACKER** 1](#_Toc169712396)

[Sommario 2](#_Toc169712397)

[Introduzione 3](#_Toc169712398)

[System Feature Utente 4](#_Toc169712399)

[Sign In 4](#_Toc169712400)

[Login 4](#_Toc169712401)

[Visualizza Posizioni 5](#_Toc169712402)

[Visualizza Posizione Attuale 5](#_Toc169712403)

[Casi d’Uso Client-Server 7](#_Toc169712404)

[System Feature Client-Server 7](#_Toc169712405)

[Public Message 7](#_Toc169712406)

[Stabilisci Connessione al Server 8](#_Toc169712407)

[Salva Topic 8](#_Toc169712408)

[Stabilisci Connessione al Database 9](#_Toc169712409)

[Requisiti Hardware 10](#_Toc169712410)

[Protocolli di Rete 11](#_Toc169712411)

[Database 11](#_Toc169712412)

[Progettazione logica 12](#_Toc169712413)

## Introduzione

La seguente relazione discuterà dei requisiti tecnici e le funzionalità di un GPS Tracker che legge e stampa e salva in database la posizione attuale dell’utente. L’utente può servirsi di una web application dedicata dopo essersi registrato per visualizzare le posizioni salvate in database e la propria posizione in tempo reale senza dover consultare il Serial Monitor dell’Arduino IDE.

**Casi d’Uso Utente**

Immagine che contiene schizzo, diagramma, bianco, cerchio

Descrizione generata automaticamente

## System Feature Utente

### Sign In

|  |  |
| --- | --- |
| ***Caso d’Uso*** | ***Data*** |
| ID | Sign In |
| DESCRIZIONE | Registrazione utente |
| ATTORI PRIMARI | Utente |
| PRECONDIZIONI | GUI dedicata, verifica input |
| EVENTI | 1. L’Utente clicca su Sign In 2. L’Utente inserisce dati 3. Il Sistema verifica i dati 4. SE 3) riesce allora inserisce dati nel database 5. L’Utente è reindirizzato alla home |
| EVENTI ALTERNATIVI | 4.1 L’Utente è invitato a  riscrivere i dati |

### Login

|  |  |
| --- | --- |
| ***Caso d’Uso*** | ***Data*** |
| ID | Login |
| DESCRIZIONE | Accesso alla web application |
| ATTORI PRIMARI | Utente |
| PRECONDIZIONI | GUI dedicata, verifica input |
| EVENTI | 1. L’Utente clicca su Login 2. L’Utente inserisce dati 3. Il Sistema verifica i dati e li confronta col database 4. SE 3) riesce allora inserisce dati nel database 5. L’Utente è reindirizzato alla GPS home |
| EVENTI ALTERNATIVI | 4.1 L’Utente è invitato a  riscrivere i dati  4.2 Se L’Utente è già loggato  ricarica la sessione |

### Visualizza Posizioni

|  |  |
| --- | --- |
| ***Caso d’Uso*** | ***Data*** |
| ID | Visualizza Posizioni |
| DESCRIZIONE | Visualizza Posizioni salvate in database |
| ATTORI PRIMARI | Utente |
| PRECONDIZIONI | GUI dedicata, connessione al database |
| EVENTI | 1. L’Utente clicca su Visualizza Posizioni 2. Il Sistema stampa tabella dal database |
| EVENTI ALTERNATIVI | * 1. Se non è connesso al   Database riporta errore |

### Visualizza Posizione Attuale

|  |  |
| --- | --- |
| ***Caso d’Uso*** | ***Data*** |
| ID | Visualizza Posizione Attuale |
| DESCRIZIONE | Visualizza Posizione Attuale tramite mappa |
| ATTORI PRIMARI | Utente |
| ATTORI SECONDARI | - |
| PRECONDIZIONI | GUI dedicata, connesione al server |
| EVENTI | 1. L’Utente clicca su Visualizza Posizione Attuale 2. Il Sistema mostra la mappa |
| EVENTI ALTERNATIVI | * 1. Se manca connessione al Server riporta errore |

## Casi d’Uso Client-Server

Immagine che contiene diagramma, schizzo, disegno, linea

Descrizione generata automaticamente

## System Feature Client-Server

### Public Message

|  |  |
| --- | --- |
| ***Caso d’Uso*** | ***Data*** |
| ID | Public Message |
| DESCRIZIONE | Public Message al Server |
| ATTORI PRIMARI | Client |
| ATTORI SECONDARI | Server |
| PRECONDIZIONI | Connessione al server |
| EVENTI | 1. Il Client riceve segnale GPS dal Sensore 2. Il Client tenta il Publishing del Topic al Server |
| EVENTI ALTERNATIVI | * 1. Se manca connessione al Server riporta errore |

### Stabilisci Connessione al Server

|  |  |
| --- | --- |
| ***Caso d’Uso*** | ***Data*** |
| ID | Stabilisci Connessione al Server |
| DESCRIZIONE | Stabilisci Connessione al Server |
| ATTORI PRIMARI | Client |
| ATTORI SECONDARI | Server |
| PRECONDIZIONI | - |
| EVENTI | 1. Il Client tenta la connessione al Server 2. Il Client si connette al Server |
| EVENTI ALTERNATIVI | * 1. Se manca connessione al Server riporta errore |

### Salva Topic

|  |  |
| --- | --- |
| ***Caso d’Uso*** | ***Data*** |
| ID | Salva Topic |
| DESCRIZIONE | Ricevi e salva Topic |
| ATTORI PRIMARI | Server |
| ATTORI SECONDARI | Database |
| PRECONDIZIONI | Connessione al Database |
| EVENTI | 1. Il Server legge il Topic dal Client 2. Il Server tenta la query al Database 3. Se 2) riesce allora salva in Database |
| EVENTI ALTERNATIVI | * 1. Se manca connessione al Database riporta errore   2. Se la query è illecita riporta errore |

### Stabilisci Connessione al Database

|  |  |
| --- | --- |
| ***Caso d’Uso*** | ***Data*** |
| ID | Stabilisci Connessione al Database |
| DESCRIZIONE | Stabilisci Connessione al Database |
| ATTORI PRIMARI | Server |
| ATTORI SECONDARI | Database |
| PRECONDIZIONI | - |
| EVENTI | 1. Il Server tenta connessione al Database 2. Se 1) riesce allora si connette |
| EVENTI ALTERNATIVI | * 1. Se manca connessione al Database riporta errore |

### Requisiti Hardware

Per la realizzazione circuitale del GPS tracker si è fatto uso dei seguenti componenti:

* Microcontrollore ESP32 Wrover B (client)
* Modulo GPS Neo 6m UBLOX (sensor)
* SBC Raspberry Pi (broker)
* 4 jumper M-M

**Schema Circuitale**

Immagine che contiene diagramma, schizzo, Disegno tecnico, bianco e nero

Descrizione generata automaticamente

**Requisiti Software**

Per l’utilizzo dell’app basta una stabile connessione ad Internet e la consultazione del file README.txt allegato alla Relazione Progettuale.

## Protocolli di Rete

Il progetto sfrutta il protocollo ISO standard MQTT per la comunicazione tra la Raspberry Pi e la ESP32. MQTT è un protocollo di messaggistica di tipo publish-subscribe appartenente alla categoria dei TCP/IP. Ottimo per progetti a banda limitata e a basso consumo energetico è stato scelto proprio per la sua semplicità d’uso e per simulare tecnicamente quello che avviene in sistemi più complessi. Infatti, in questo progetto è stato utilizzato un solo GPS tracker per ovvie ragioni tecniche, ma potenzialmente si potrebbero sviluppare una moltitudine di dispositivi che fanno da client in grado di lavorare contemporaneamente grazie all’opera di “smistamento” dei messaggi ricevuti dal broker. Inoltre, si serve del protocollo HTTP per la connessione ad Internet tipica delle applicazioni client-server.

## Database

Il progetto sfrutta come gestore del database il software XAMPP che permette l’installazione di un server Apache HTTP, di un database server MariaDB e del linguaggio di programmazione PHP. Di qui si tratterà nello specifico della programmazione del suddetto database.

**Progettazione concettuale**

**Diagramma ER**

Immagine che contiene diagramma, Disegno tecnico, Piano, schizzo

Descrizione generata automaticamente

## Progettazione logica

user(**ID**, admin, password).

position(**ID,** lat, lg).

## Possibili sviluppi futuri

Il nostro GPS-TRACKER è stato studiato per utilizzarlo insieme alla consultazione dell’apposita web application o tramite il monitor del Raspberry Pi, in alternativa può essere implementato al circuito un qualsiasi monitor OLED per la visualizzazione della mappa e/o dei dati satellitari. Inoltre, è studiato e collaudato per funzionare collegato ad un PC, è quindi consigliato sviluppare in futuro una connessione wireless alla UART del ESP32 per permetterne la mobilità e l’indipendenza dal PC. Si può pensare anche di inscatolare il GPS per renderne la fruibilità migliore.