# Introduzione

Informazioni sul progetto

Il progetto Tankino è stato realizzato da Adriano Chiriacò, Nicholas Pigni, Mosè Ferrazini e Michele Tomyslak alla Scuola Arti e Mestieri a Trevano durante la lezione del modulo 306.

I docenti responsabili sono:

- Luca Muggiasca (luca.muggiasca@edu.ti.ch)

- Geo Petrini (geo.petrini@edu.ti.ch)

La data d'inizio del progetto è il 17.01.2020 mentre la fine è prevista per il 22.05.2020.

## Abstract

*In questo documento si possono trovare tutte le informazioni che riguardano lo sviluppo del progetto di Tankino. Più nel dettaglio si possono trovare informazioni su come abbiamo gestito la progettazione del nostro prodotto, i mezzi utilizzati, i requisiti da rispettare, come è avvenuta l’effettiva realizzazione, il design dell’applicazione, i vari protocolli di comunicazione e i test eseguiti.*

## Scopo

Lo scopo del progetto Tankino è di realizzare la base di un carro armato con dei motori elettrici. Questi motori dovranno essere controllati da un arduino in modo da poter muovere il carro armato a piacere. Inoltre l’arduino riceverà dei dati da un dispositivo mobile tramite bluetooth in modo da poter controllare il carro armato a distanza. Per poter controllare il carro armato dal dispositivo mobile è inoltre richiesto di sviluppare un applicazione da cui sia possibile controllare il carro armato con dei comandi semplici e intuitivi.

# Analisi

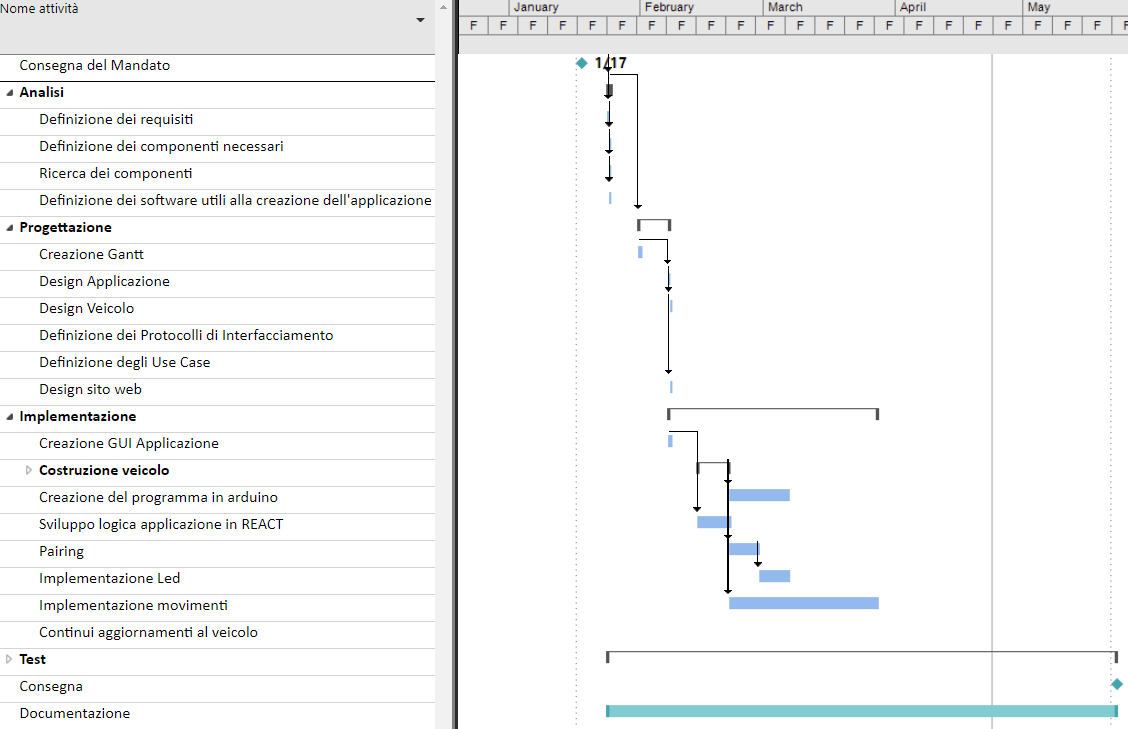
## Analisi e specifica dei requisiti

|  |  |  |  |
| --- | --- | --- | --- |
| Id requisito | Requisito | Priorità | Versione |
| REQ-001 | Avere un’applicazione mobile | 1 | 1.0 |
| REQ-002 | Avere un arduino funzionante | 1 | 1.0 |
| REQ-003 | L’applicazione deve permettere di pilotare il veicolo | 2 | 1.0 |
| REQ-004 | Essere in grado di fare un pairing bluetooth | 2 | 1.0 |
| REQ-005 | Il veicolo deve essere in grado di muoversi avanti e indietro | 2 | 1.0 |
| REQ-006 | Il veicolo deve essere in grado di curvare | 3 | 1.0 |
| REQ-007 | Il veicolo deve avere delle luci di controllo, gestibili da applicazione | 4 | 1.0 |
| REQ-008 | Avere un sito dove scaricare le applicazioni | 3 | 1.0 |
| REQ-009 | Il veicolo deve essere pilotato in tempo reale con la minor latenza possibile (misurata). | 3 | 1.0 |
| REQ-010 | La velocità deve essere variabile | 4 | 1.0 |

## Use case

I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto.

# Pianificazione

Per fare la pianificazione abbiamo creato un Gantt cercando di inserire all’interno tutte le attività che ci siamo posti, e suddividendocele.

## Analisi dei mezzi

I mezzi da noi utilizzati per quanto riguarda la parte software e quindi la programmazione sono stati Android Studio e Arduino IDE. Per eseguire la creazione del Gantt abbiamo utilizzato un prodotto del pacchetto Office chiamato “Project”

Mentre per quanto riguarda l’hardware ci siamo provvisti di due step motor “nema 8”, di un powerbank “Aukey” da 10’000mah e due controller per gli step motor “L298N”.

Il programma per controllare il Tankino bisognerà avviarlo su un qualsiasi dispositivo android.

# Progettazione

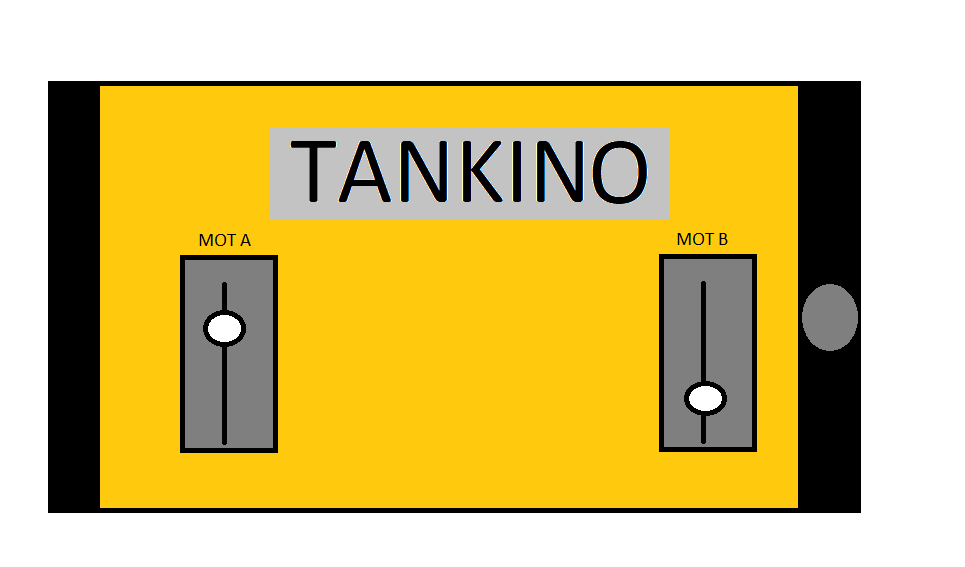
La progettazione del nostro prodotto è variata tanto nel tempo, ma dopo varie considerazioni siamo arrivati a questa conclusione. La base fisica dal Tankino verrà realizzata in legno. Le ruote, prodotte in plexiglass saranno collegate tra loro con dei cingoli, i cingoli che sono di marca “lego”. Per quanto riguarda l’alimentazione utilizzeremo un powerbank con due uscite: una da 1v e 1a e una da 1v e 2a. Nella prima ci collegheremo l’arduino mentre la seconda servirà ad alimentare i motori. All’arduino ci sarà anche collegato un ricettore bluetooth che servirà a controllare il Tankino tramite l’applicazione per smartphone. Inoltre sempre all’arduino ci saranno collegati i due controller che permetteranno all’arduino di gestire gli step motor. Per spiegare meglio tutta questa parte di elettronica abbiamo deciso di sviluppare uno schema elettrico.

Un’altra cosa a cui abbiamo pensato è il protocollo di comunicazione tra l’applicazione all’arduino tramite bluetooth. La nostra idea è questa:

**Protocollo di comunicazione**

|  |  |  |
| --- | --- | --- |
| **Messaggio** | **Nome** | **Descrizione** |
| MSB = 0  0XXX XXXX | Cingolo destro | Il cingolo destro si muove in base alla percentuale data (da -100% a +100%)  L’MSB determina che il cingolo che si deve muovere è il destro, il resto è il valore passato che è compreso nel range [-64;63] (oppure [0-127]).  Il cingolo si muove in senso avanti se il valore è positivo e indietro se il valore è negativo. |
| MSB = 1  1XXX XXXX | Cingolo sinistro | L’MSB determina che il cingolo che si deve muovere è il sinistro, il resto è il valore passato che è compreso nel range [-64;63] (oppure [0-127]).  Il cingolo si muove in senso avanti se il valore è positivo e indietro se il valore è negativo. |

## Design

L’interfaccia dell’applcazione per mobile sarà molto semplice, sarà basata su due slider che permetteranno di gestire in modo autonomo i singoli motori. Ovviamente portando lo slider al massimo il motore raggiungerà la velocità massima e portandolo al minimo il motore sarà fermo. Questo è il nostro progetto per quanto riguarda il design dell’applicazione:

Per quanto invece riguarda il design del tankino vero e proprio sarà formato da due tavole di legno poste una sopra l’altra nel quale centralmente sarà posto un buco per far passare i cavi da sopra a sotto. Le ruote saranno realizzate in plexiglass e con dimensioni uguali tra loro.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

|  |  |
| --- | --- |
| Test Case | TC-001 |
| Nome | Avere un’applicazione mobile android |
| Riferimento | REQ-001 |
| Descrizione | Test case per verificare di avere un’applicazione android funzionante. |
| Prerequisiti | Avere un dispositivo android. |
| Procedura | Installare l’applicazione tramite l’apk su un dispositivo android e aprirla. |
| Risultati attesi | L’applicazione viene installata e aperta senza problemi. |

|  |  |
| --- | --- |
| Test Case | TC-002 |
| Nome | Avere un’arduino funzionante |
| Riferimento | REQ-002 |
| Descrizione | Test case per verificare di avere un’arduino funzionante. |
| Prerequisiti | Avere un pc a disposizione. |
| Procedura | Collegare l’arduino al computer e provare a caricarci un programma. |
| Risultati attesi | L’arduino viene individuato dal computer e esegue il programma correttamente. |

|  |  |
| --- | --- |
| Test Case | TC-003 |
| Nome | Test guidabilità veicolo |
| Riferimento | REQ-003 |
| Descrizione | Test case se L’applicazione permette di pilotare il veicolo |
| Prerequisiti | Essere connessi via bluetooth al tankino |
| Procedura | Provare a muovere entrambe gli slider sull’applicazione |
| Risultati attesi | Il tankino dovrebbe muoversi. |

|  |  |
| --- | --- |
| Test Case | TC-004 |
| Nome | Test guidabilità veicolo |
| Riferimento | REQ-004 |
| Descrizione | Test case se L’applicazione permette di pilotare il veicolo |
| Prerequisiti | Essere connessi via bluetooth al tankino |
| Procedura | Provare a muovere entrambe gli slider sull’applicazione |
| Risultati attesi | Il tankino dovrebbe muoversi. |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Questo è il nostro

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Bibliografia per articoli di riviste

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo,

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

Esempio:

* <http://standards.ieee.org/guides/style/section7.html>, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o Qdc
* Prodotto
* …