**STATO AVANZAMENTO TEO – ottobre 2016**

**PULSANTI CAPPELLO **

* I pulsanti erano già presenti sul cappello come previsto nella versione 1 del robot.
* Ho previsto il loro impiego e l’hardware presenta 6 pulsanti, facilmente collegabili ed usabili.
* Mancano da realizzare i cavi (calcolare bene la lunghezza!)
* NB: pulsanti non testati ma sui pin del PCB arriva la giusta tensione (misurata), quindi devono funzionare.

**Hardware: 90% 🡪 tutto ok tranne cavi**

**Software: 40% 🡪 nessun software realizzato per la gestione (si adatta quello vecchio)**

**SENSORI PIR, SONAR E IR **

* Ho previsto il loro utilizzo per applicazioni future con lo scopo di conoscere la posizione del bambino rispetto a Teo e a che distanza esso si trova. (fondamentale per mantenere le corrette distanze sociali – vedi tesi)
* Esistono sul modulo componenti attacchi per i 3 pir, per il sensore IR e per il sonar.
* La disposizione fisica dei sensori IR e Sonar è alla base, in posizione frontale, in quanto nella scocca nera essi erano già montati in quella posizione (scocca forata e sensori inseriti); invece i sensori pir non hanno ancora una disposizione fissa e posso essere posizionati dove meglio si desidera.
* NB: sensori non testati ma sui pin del PCB arriva la giusta tensione (misurata), quindi devono funzionare.

**Hardware: 60% 🡪 bisogna disporre i PIR (analisi comportamentale Teo e utilizzo sensori, trovare posizione ideale, lavori di montaggio) e realizzare i cavi.**

**Software: 0% 🡪 nessun software realizzato per la gestione**

**SENSORE ACCELEROMETRO/GIROSCOPIO **

* Su Teo2 sfrutto il sensore giroscopio per capire se viene ribaltato.
* Il sensore funziona (stranamente) molto bene.
* Unica cosa da fare è trovare il modo di fissarlo stabilmente in posizione verticale sotto al cappello.

**Hardware: 90% 🡪 bisogna trovare il modo di fissare il sensore sotto al cappello e realizzare i cavi della giusta lunghezza.**

**Software: 100% 🡪 funzione di gestione già realizzata, testata e funzionante.**

**STRISCIA LED E LED MATRIX **

* Le strisce led e la matrice led funzionano bene.
* Ho realizzato funzioni che gestiscono la faccia e le animazioni delle strisce in base allo stato d’animo del robot.

**Hardware: 100% 🡪 tutto ok**

**Software: 100% 🡪 tutto realizzato, al massimo è da ottimizzare**

**MOTORI**

* uso la libreria Triskar e passo i valori come nel Teo vecchio
* il comportamento non è quello atteso, la velocità di movimento è molto superiore alle aspettative
* bisogna fare un check hardware perché probabilmente è lì il problema
* altro check sui singoli motori, vedere se qualcuno riceve la giusta alimentazione, se tutti i pin dei driver sono collegati correttamente
* esiste un pin dei driver che nella versione vecchia di Teo, per due soli driver su tre, era collegata a 5V; in questa versione ho realizzato una pista che porta i 5V ma, non capendo se fosse necessario collegarli o meno, la pista non è collegata (ho proprio modellizzato così il PCB); nel caso serva, per farlo basta saldare una goccia di stagno in prossimità dell’interruzione.

**Hardware: 60 - 80% 🡪 bisogna fare un check che sia stato realizzato correttamente perché il problema descritto sopra può essere dovuto a questo.**

**Software: 100% 🡪 già realizzate le funzioni che gestiscono il movimento sfruttando la libreria triskar**

**MICROFONI **

* Il problema principale dei microfoni resta, a mio parere, la scarsa qualità degli stessi.
* Algoritmo di sound localization “funziona” con i microfoni disposti sul tavolo, ma una volta montati presenta delle grosse limitazioni.
* Algoritmo è sperimentale e si basa su mie osservazioni e considerazioni personali, che possono essere smentite e superate.
* Il grosso problema è la ricezione dei valori, che a volte sono molto sballati: si sono verificate situazioni in cui, in condizione di silenzio, un microfono riceve valori stazionari che sono anche 50 volte inferiori agli altri due microfoni, oppure nel minore dei casi, riceve solo una serie di valori ogni tanto sballati.
* Situazioni anomale di questo tipo ostacolano lo sviluppo dell’algoritmo.

**Hardware: 70% 🡪 bisogna capire come fissare i mic al cappello (rifare cappello?), rifare i cavi (lunghi abbastanza!), capire se è necessario acquistare mic di qualità superiore**

**Software: 80% 🡪 software di lettura valore dai mic esiste, software che implementa l’algoritmo di sound localization esiste ma presenta le problematiche descritte sopra**

**SENSORE DI CONTATTO **

* Il riconoscimento del contatto avviene in maniera abbastanza precisa quando il sensore non è montato su Teo.
* Il problema si presenta quando i sensori sono disposti nelle tasche del robot. Qui i valori di lettura degli FSR subiscono una distorsione che dipende da due fattori: il primo è la forma arrotondata del robot (superabile, in quanto basta ricalibrare il valore iniziale), la seconda invece è legata alla rotazione/movimento involontario della parte terminale sensore. Cerco di spiegare meglio: i sensori FSR, che siano strisce o quadrati, hanno una parte finale sottile che termina con due pin e il problema è che se anche questa parte finale subisce delle torsioni, viene toccata o altro, l’intero sensore rileva che c’è stata una pressione e la classifica alla pari di un colpo violento. Quindi, quando si inserisce il sensore nella tasca, capita che questa parte terminale dell’fsr viene mossa, toccata, ruotata, semplicemente non rimane dritta e il valore iniziale del sensore non è al pari degli altri.
* La soluzione consiste o nel trovare il modo mantenere la parte terminale del sensore dritta (difficile), o di sviluppare l’algoritmo in modo che non parta da una soglia fissa (come teoricamente dovrebbe essere sviluppato). Il problema alla seconda soluzione è che se la distorsione della parte finale del sensore è eccessiva (ad esempio senza toccare nulla il sensore rileva 800 su 1023), è difficile distinguere il range numerico di contatto.

**Hardware: 70% 🡪 bisogna trovare il modo di risolvere descritto sopra**

**Software: 50% 🡪 va rifatto: algoritmo che si basa sulla media dei valori a riposo VS calibrando i sensori rispetto dopo essere stati inseriti su Teo e determinando fasce di tocco (colpo, carezza, ecc). Il codice di lettura FSR presente**

**MODULO BLUETOOTH TEO **

* Comunicazione BT è point to point.
* Il BT su Teo comunica con il telecomando e non può comunicare anche con il PC.
* In teoria Teo ha bisogno di comunicare con il PC perché quest’ultimo gestisce la riproduzione dei messaggi vocali sulla cassa BT disposta su Teo. (Problema non si pone se elimino il telecomando, quindi con una comunicazione solo con PC/Telefono)
* Bisogna risolvere questa cosa, ci sono quattro modi: 1) inserisco un nuovo modulo BT (soluzione facile ma non ottimizzante); 2) cerco un modulo BT (possibilmente HC) che consente di realizzare una rete di device. 3) Cambio tutti i device e uso BLE che funziona; 4) elimino telecomando
* 🡪 devo realizzare una rete di device BT in cui BT\_TEO è al centro e comunica sia col Telecomando che con il PC

**NUOVO: rifare la rete di device/aggiungere nuovo BT**

**Riconfigurare la rete e la comunicazione – gestire riproduzione audio**

**BATTERIE LIPO **

* Uso due batterie, una 12V per alimentare direttamente i motori e una 7V per alimentare arduino e tutti i componenti.
* Bisognerebbe fare un calcolo e capire se nel peggiore dei casi (tutti i componenti previsti contemporaneamente in funzione) le batterie riescono a gestire tutta la sensoristica.
* Bisognerebbe trovare anche altre nuove batterie ed altri modi per gestire tutto.

**Hardware: 100% 🡪 Teo è stato testato con le batterie Lipo e funziona per alimentare contemporaneamente Arduino, Led Strips (x2), Led Matrix, Accelerometro, Motori – E**

**ncoder – Driver.**

**RICONOSCIMENTO VOCALE EASY VR **

* E’ presente una shield per implementare la funzionalità di riconoscimento vocale, montata sull’hardware superiore
* Non è stato sviluppato alcun codice, in questo senso l’attività parte da 0

**Hardware: 80% 🡪 hardware realizzato secondo i vari esempi di funzionamento ricercati nel web; non è stato testato il suo funzionamento.**

**Software: 0 🡪 bisogna fare analisi e test preliminari per capire se l’hardware di base è stato realizzato correttamente; in seguito trovare il modo di far funzionare il componente**

**TELECOMANDO**

* Del telecomando è stato previsto il design basandosi sul giusto compromesso tra UX e scalabilità in relazione alle attività di Teo (vedi tesi).
* In base alle scelte di UX e design, ho sviluppato il PCB, ho fatto i fori su di esso e ho realizzato il modello 3D.
* Bisogna realizzare fisicamente il telecomando, ovvero: saldare gli strip (come previsto al primo punto), realizzare le connessioni interne tra i vari elementi, far stampare il modello 3D del telecomando, montare i vari elementi (tasti, joystick, display OLED, BT, Arduino, ecc – vedi tesi).
* Sviluppare lo sketch Arduino che gestisce l’interfaccia utente del display OLED e invia i codici come previsto.
* NOTA: Se si decidesse di avere una diversa UX, di disporre in modo diverso i tasti o di apportare qualunque altra modifica hardware, basta realizzare un nuovo PCB   
  🡪 hardware da me realizzato non è vincolante

**PROBLEMATICHE LEGATE AI CAVI**

Alcuni cavi di collegamento degli elementi disposti nella parte superiore di Teo vanno rifatti perché non sono lunghi abbastanza.

**Suggerimenti PROGETTI**

**TELECOMANDO**

*Hardware (60%), Software (40%)*

* Realizzare hardware (come è stato previsto)
* Stampa 3D
* Programmare display (come è stato previsto)
* Gestire comunicazione da seriale Bluetooth (invio codice - ricezione)
* Controllo di Teo (joystick) e delle attività

**TELECOMANDO DA APP**

*Hardware (10%), Software (90%)*

* App per smartphone Android (ibrida o nativa)
* Gestire Teo da Telefono (telecomando virtuale)
* Grafica intuitiva e user friendly
* Gestire comunicazione seriale Bluetooth (invio codice – ricezione)
* Controllo di Teo (joystick virtuale) e delle attività

**INTERFACCIA PC GESTIONE AUDIO**

* App per gestire la comunicazione audio

Solo funzioni in base ad una ristrutturazione totale del codice

**RETE DI DEVICE BT**

Creare una rete di device BT: telecomando, Teo, Pc, Casse BT

**SENSORI DI CONTATTO**

Dato il corpo di Teo e date le strisce e i sensori fsr, gestire il contatto e dare in output un valore di un enum che rappresenta l’intensità del contatto.

**CAPPELLO E MICROFONI**

*Hardware (50%), Software (50%)*

* Ricostruire il cappello di Teo (modello, stampa 3D, rivestimento)
* Disporre 3 microfoni attorno al cappello e i pulsanti sopra
* Sviluppare un sistema di sound localization (identificare la posizione di un suono)

**ATTIVITA’ IDLE**

Pensare e sviluppare l’attività di idle

**NASCONDINO**

Implementare l’attività nascondino con i sensori disponibili ed eventualmente aggiungerne altri