

## I.I.S. "EINSTEIN" - Vimercate (MB)



## Progetto

**AID** (Arduno with Itegration for Disabled)

### Autori

Lorenzo     **LUCCA**     4D Info

Hamza     **HADDOU**     4D Info

**Tutor**     Paolo Di Bella

## INDICE

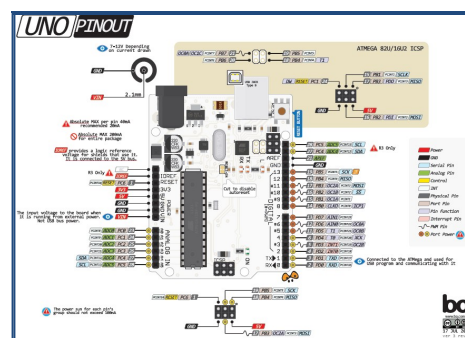
Descrizione	pg. 3
1) Introduzione – Problematiche della ricerca	pg. 3
2) Progettazione Hardware e Software	pg. 5
a) Caratteri generali	pg. 5
b) Diagramma di GANNT	pg. 6
c) Processi del sistema	pg. 6
d) Definizione dei <i>task</i> principali	pg. 7
3) Esecuzione e realizzazione	pg. 7
a) Cablaggio hardware	pg. 7
b) Interfaccia utente	pg. 8
c) Sviluppo del software	pg. 8
d) Specifiche del SW di massima	pg. 9
e) Flow-Chart	pg. 9
f) Progettazione di dettaglio del SW – Moduli JAVA	pg. 10
4) Conclusioni - Collaudo	pg. 12
5) Conclusioni - Miglioramenti, nuove alternative e possibili utilizzi	pg. 13
6) Conclusioni - Ringraziamenti	pg. 13

**Abstract:** Il progetto è stato sviluppato con l'obiettivo di realizzare un sistema integrato basato sulla piattaforma Arduino rivolto a studenti degli Istituti Tecnici e Professionali che presentano disabilità fisiche e cognitive. Esso consente a tali studenti di poter svolgere le medesime esperienze di laboratorio di Elettronica proposte alla classe di riferimento in modo autonomo. Il sistema, infatti, prevede una interfaccia grafica ad icone con modulo *touchscreen*, un ambiente di sviluppo del software semplificato, connessioni hardware facilitate e cablaggio guidato. Lo studente sarà in grado, attraverso l'uso di questa nuova piattaforma HW/SW, di realizzare le esercitazioni di laboratorio programmate per l'attività didattica in autonomia. Il progetto ricalca fedelmente l'ottica di realizzare un progetto *dalla classe per la classe* con l'obiettivo di una maggiore inclusività di studenti che presentano problemi psicofisici. Il sistema è scalabile, aggiornabile e modificabile per ulteriori applicazioni.

**Descrizione:** La nostra idea ha la finalità di progettare e realizzare un sistema integrato didattico trasversale utilizzabile nei laboratori di Sistemi/TPS/Elettronica degli Istituti Tecnici e Professionali ed anche in quelli di Tecnologia delle Scuole Secondarie di Primo Grado. Il progetto prevede la costruzione di un prototipo funzionante basato sul microcontrollore Arduino che consente anche a studenti con disabilità visive, cognitive e di motricità di realizzare autonomamente



Il microcontrollore Arduino



Il pin-out di Arduino

le medesime esperienze di laboratorio proposte al gruppo classe; tale realizzazione sarà facilitata dal supporto software ed hardware implementato nella nostra soluzione. Gli studenti, quindi, anche senza il supporto degli insegnanti di sostegno, saranno in grado di assemblare e collaudare il medesimo circuito con l'ausilio di interfacce grafiche iconiche su *touchscreen* ed un hardware semplificato che consentirà loro di completare l'esperienza proposta con un ampio livello di autonomia.

## 1) INTRODUZIONE – Problematiche della ricerca

Il progetto si rivolge in modo particolare agli studenti che presentano disabilità psico-fisiche e di apprendimento e che, quindi, necessitano di strategie e modalità di insegnamento adeguate alle loro caratteristiche e, in particolare modo nelle discipline applicative che rivestono grande importanza nel piano di studi del triennio ITIS; fino ad oggi le esperienze di laboratorio di elettronica e di materie tecniche affini sono state eseguite con il supporto dei compagni di classe, degli insegnanti di sostegno e degli insegnanti di disciplina che hanno cercato di

semplificare il più possibile tali esperienze, talvolta tecnicamente complesse, in modo da consentire anche a questi studenti una completa realizzazione come i loro compagni di classe.

Si è pensato quindi, in un'ottica di una maggiore inclusione, di progettare un sistema integrato basato sulla *board* Arduino che potesse semplificare la realizzazione hardware e software dell'esercitazione proposta in modo tale da consentire allo studente disabile di poter completare in modo autonomo e quindi gratificante, la stessa esperienza di laboratorio degli altri alunni della classe.

Tale sistema è basato su di una semplificazione dei processi hardware e software da sviluppare ma ugualmente in grado di fornire allo studente una visione globale della singola esperienza e di raggiungere l'obiettivo finale.

La realizzazione delle esperienze di laboratorio hanno sia la valenza di aumentare l'autostima dello studente che riesce a raggiungere un obiettivo ben definito e sia quello di un miglioramento sul piano didattico sperimentato sul campo in quanto rappresenta un'esperienza stimolante, multidisciplinare e di interazione con elementi tecnici presenti nella realtà e quindi anche molto utili per l'acquisizione di nuove competenze.

In modo particolare la nostra applicazione si rivolge a studenti che presentano disturbi cognitivi, visivi e di motricità fine; in questi casi si rilevano spesso *deficit* di precisione, coordinazione e di sviluppo di un percorso logico e ciò, ovviamente, rappresenta un forte limite anche nell'utilizzo delle periferiche standard esistenti (tastiera, mouse, cablaggi di circuiti elettronici); con l'obiettivo di tenere conto di tali specifiche difficoltà, nella nostra soluzione si è optato per l'utilizzo del *touch screen*, ossia uno schermo incorporato nel monitor e sensibile al tatto che elimina



**Interazione con *touch-screen***



**Il nostro touchscreen ASUS Transformer**

completamente la mediazione di una periferica per operare col computer, rendendo l'interazione più intuitiva e quindi semplice anche sul piano cognitivo. Le tecnologie *touchscreen* sono in grado di fornire un profondo ausilio nel settore *education* anche per superare le difficoltà di apprendimento. Attraverso questa tecnologia si traggono benefici per uno sviluppo e una comunicazione più efficace e risulta anche di valido aiuto agli studenti che hanno difficoltà di attenzione o di lettura che vengono supportati, come nel caso del nostro progetto, da oggetti visuali e dalla facilità di interazione. Le esperienze proposte nel corso di Elettronica rappresentano anche una modalità con cui si favorisce lo sviluppo didattico in

chiave tecnologica ma anche relazionale con il gruppo classe; inoltre, grazie alla notevole adattabilità della tecnologia, è possibile "cucire" gli ausili esistenti alle esigenze del singolo studente in base alle sue caratteristiche per creare concretamente valide opportunità per potenziarne le abilità.

Per armonizzare entrambe le necessità cioè, un hardware semplificato per consentire dei movimenti più semplici per il cablaggio ed un software che vada incontro alle difficoltà di apprendimento, si è scelto un PC con interfaccia touchscreen, un cablaggio guidato da PC basato su cavi elettrici di maggiori dimensioni e *jack* a banana che riduce al minimo gli errori di collegamento fisico ed un software che si autogenera in base alle scelte grafiche operate dallo studente, in modo tale da consentirgli autonomamente ed in modo trasparente di superare le difficoltà nello scrivere programmi applicativi. Le board di Arduino e delle periferiche (trasduttori, display LCD, attuatori) saranno contenute in box trasparenti per verificare visivamente tutte le operazioni che si stanno svolgendo.

## 2) PROGETTAZIONE HARDWARE E SOFTWARE

### a) Caratteri generali

L'idea di fondo è di realizzare un prototipo completamente funzionante e pronto da usare nelle classi individuate. La scelta iniziale è ricaduta sulla piattaforma Arduino per ragioni di costo, reperibilità, vastissima gamma di *shields* compatibili, un IDE completamente gratuito e *sketch* software già disponibili sul web e, quindi, con la concreta possibilità di rendere scalabile il nostro sistema a seconda dell'applicazione scelta.

Per la realizzazione hardware si è partiti dall'obiettivo finale per suddividere successivamente il progetto in sottoprogetti per ciascuno dei quali si è scelta la soluzione tecnica più agevole, economica, ed efficiente svolgendo un confronto tra le eventuali soluzioni possibili in commercio.

Per la parte software si è scelta una metodologia *top-down*; abbiamo formulato inizialmente una visione complessiva del sistema attraverso le specifiche generali che successivamente è stata elaborata in modo più dettagliato fino a giungere alla codifica in Java del programma.

Tutte le fasi di progettazione e di realizzazione sono state codificate da un diagramma temporale di GANNT al fine di pianificare e coordinare le singole attività del progetto fornendo una chiara illustrazione dello stato d'avanzamento.

b) Diagramma di GANNT

ATTIVITA'	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
Brainstorming docenti + gruppo classe	■														
Definizione obiettivi generali		■													
Progettazione tecnica dei box			■												
Scelta tecnologica per il cablaggio			■												
Realizzazione dei box necessari			■												
Definizione delle esperienze da svolgere da parte condivise con il CdC			■												
Definizione specifiche SW				■											
Definizione scelta tecnologica per interazione					■	■	■								
Flow-chart SW					■	■	■								
Programmazione Java dell'interfaccia grafica							■	■	■	■	■				
Programmazione dello sketch di Arduino									■	■	■	■			
Comunicazione sw PC-Arduino										■	■	■			
Debugging del SW								■	■	■	■	■	■	■	■
Collaudo generale														■	■
Debug del sistema integrato				■	■	■	■	■	■	■	■	■	■	■	■
Applicazione in laboratorio															■

c) Processi del sistema - Generale

1. Accensione PC ed accesso al pannello operatore *touch* che presenta allo studente una interfaccia grafica basata su un sistema di menu ad icone *parlanti*
2. Il menù principale prevede 4 tipologie di esperienze (vettori LED, trasduttori, un attuatore ad es. motore c.c., display LCD) tra cui scegliere l'esperienza specifica da realizzare
3. Una volta effettuata la scelta si richiede la selezione delle variabili a seconda dell'opzione eseguita (ad esempio tipo di sequenza LED, messaggio per il display LCD, etc.)
4. In base alle selezioni dei precedenti menù, il software prevede la modalità "guida operativa" che guiderà lo studente in modo grafico/parlante *step-by-step* alla connessione del circuito; per ciascuna di operazione di cablaggio, che sarà comunque facilitato, verrà visualizzato quali connettori collegare (con descrizione) in modo da ridurre al minimo gli errori e condurlo verso l'esperienza didattica funzionante
5. Lo *sketch* di Arduino, auto programmato ed auto compilato, verrà inviato dal PC alla *board* con le variabili scelte già assegnate

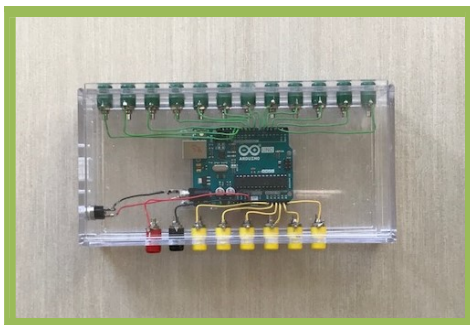
#### d) Definizione dei *task* principali

- 1) Scelta dell'icona relativa all'applicazione su schermo *touch* (LED, display LCD, trasduttori, attuatori)
- 2) Apertura di sottomenù per la scelta specifica del progetto che lo studente intende realizzare
- 3) Scelta definita tra un numero limitato di realizzazioni circuitali per ciascun applicazione
- 4) Inserimento dati variabili relativi all'esperienza (messaggio, sequenza LED, tipologia di grandezza da misurare, etc.)
- 5) L'icona scelta deve essere collegata al relativo sketch di Arduino che, in modo automatico, viene generato e lo trasferisce in memoria
- 6) Board Arduino cablata e montata su *breadboard* e incapsulata in un contenitore con cablaggio esterno facilitato e guidato. Idem per dispositivi esterni
- 7) Eventuale realizzazione stampante 3D dei box per Arduino e dispositivi esterni
- 8) Studio del cablaggio facilitato e guidato tra board e dispositivi
- 9) Collaudo e verifica in classe delle esperienze previste

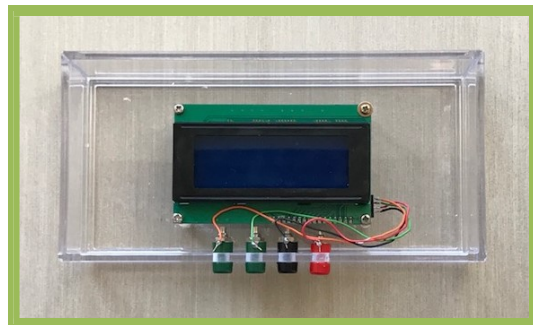
### 3) ESECUZIONE E REALIZZAZIONE

#### a) Cablaggio hardware

Da una valutazione e uno screening del mercato abbiamo rilevato che non esistono box che permettano l'estensione all'esterno dei contatti di Arduino. Si è proceduto quindi al disegno ed alla creazione di *box* in plexiglas trasparenti, in modo da non perdere mai di vista l'oggetto del problema, aggiungendo un sistema di boccole esterne che replicano i contatti fisici interni. In questo modo i collegamenti hardware saranno effettuati, anziché con fili molto sottili e difficili da inserire nei relativi fori, con *jack* di grosse dimensioni con connettori a banana che garantiscono una bassa possibilità di errore.



**Box Arduino**



**Box Display LCD I2C**

## b) **Interfaccia utente**

Per la scelta dell'interazione con lo studente si sono analizzate le diverse tipologie adatte ai casi in esame; mouse, joystick, microswitch. Tutte queste soluzioni, seppure migliorative rispetto ad un uso della tastiera come nelle applicazioni tradizionali, ci sono apparse poco consone e con un valore aggiunto non sempre adeguato all'applicazione. Si voleva raggiungere una semplicità di uso dal punto di vista pratico con relativa grafica parlante che potesse aiutare lo studente nella operatività e nella scelta. Si è optato per uno schermo *touchscreen* che con bottoni grafici a video di grandi dimensioni e relative icone associate rendesse quasi impossibile sbagliare l'operazione da svolgere.

## c) **Sviluppo del SW**

Come anticipato nell'introduzione lo sviluppo del software, nonostante sia parte integrante del progetto, non può essere svolto autonomamente dallo studente a causa delle sue difficoltà. Ci siamo, quindi, proposti di progettare una modalità che potesse rendere autonomo, gratificante e trasparente lo sviluppo della parte SW ed, allo stesso tempo, molto semplificata in quanto il linguaggio di programmazione di Arduino è di tipo logico-strutturato.

La soluzione finale è risultata essere quella di utilizzare *sketch* predefiniti dove i valori delle variabili impiegate (ad esempio: numero dei led, scelta della sequenza di accensione, messaggio da scrivere, etc) vengono passate dalla nostra applicazione Java direttamente al programma Arduino in modo automatico in seguito alla scelta effettuata sul *touch*. Una volta che il programma è completato viene compilato in locale e trasmesso in modalità seriale ad Arduino. Questa soluzione non prevede l'intervento dello studente direttamente sul programma, ma gli consente di ottenere al termine dell'esperienza un sistema funzionante come il resto della classe senza alcun supporto didattico.

Abbiamo analizzato le possibilità tecniche per la comunicazione PC/Arduino:

- dotare di uno *shield* Ethernet Arduino (in modo da ottenere un web server)
- compilare nell'ambiente IDE direttamente lo *sketch* ed inviarlo alla RAM di Arduino (seriale via USB)
- passaggio solo dei dati delle variabili via seriale a 19200

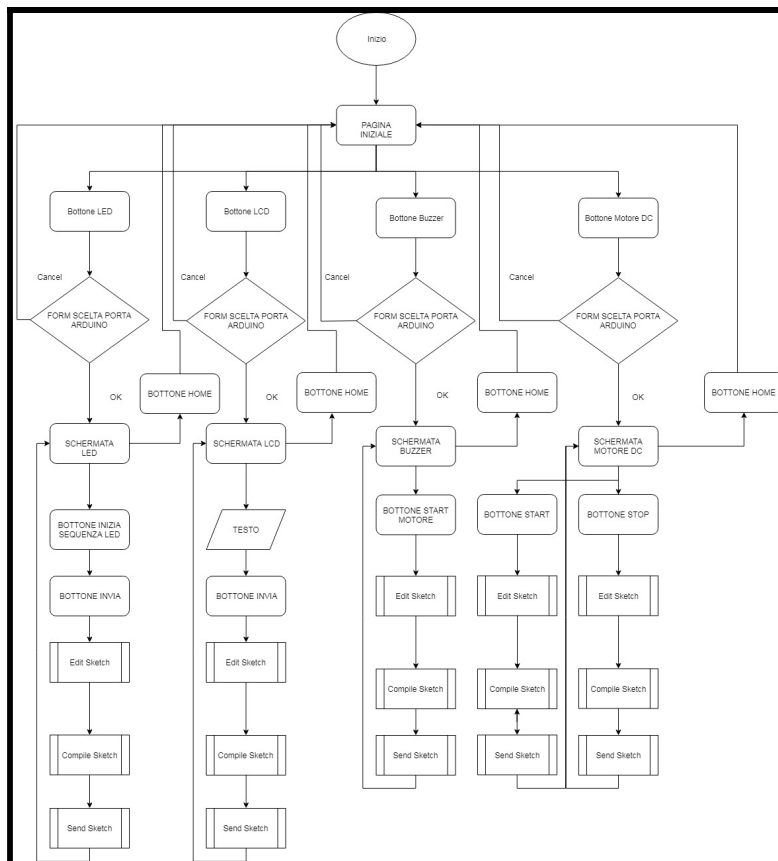
Abbiamo optato per la soluzione di utilizzare il modulo SerialManager di Arduino che consente di avere tutti i vantaggi sopra descritti.



## d) Specifiche di massima del software

- A. Pannello operatore *touch* per consentire un accesso diretto all'interfaccia
- B. Interfaccia grafica basata su un sistema di menù ad icone *parlanti* a 4 livelli fino alla definizione precisa dell'esperienza da realizzare
- C. Il sistema di menù prevedere 4 tipologie di esperienze (vettori LED, trasduttori, un attuatore ad es. motore c.c., display LCD)
- D. L'utente potrà scegliere l'assegnazione di alcune variabili (ad es. quali led e quale sequenza attivare, cosa scrivere nel display LCD, quale velocità in PWM del motore c.c.)
- E. Una volta che l'utente ha scelto, in base alle selezioni nei precedenti menù, deve essere guidato alla realizzazione pratica (sempre in modalità grafica dal PC)
- F. Per ciascuna operazione di cablaggio verrà indicato da PC quali connettori collegare (con descrizione) in modo da ridurre al minimo gli errori e condurre una esperienza di laboratorio funzionante
- G. lo *sketch* di Arduino sarà auto compilato ed inviato dal PC alla *board* con le variabili già assegnate
- H. rendere il sistema indipendente rispetto al PC e al sistema operativo utilizzato

## e) Flow Chart



Flow chart SW - Generale

Il flow chart rappresenta il menù completo di scelta dell'esperienza da svolgere che appare sul *touchscreen*. A titolo esemplificativo si sono scelte 4 esperienze; display LCD I2C, trasduttore *buzzer*, attuatore motore c.c. e vettore LED. Per ciascuna della scelta effettuata verrà creato ed inviato in modo automatico il relativo sketch ad Arduino.

## f) Progettazione di dettaglio del SW – Moduli JAVA

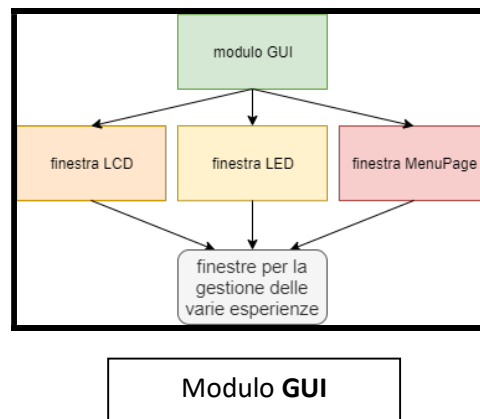
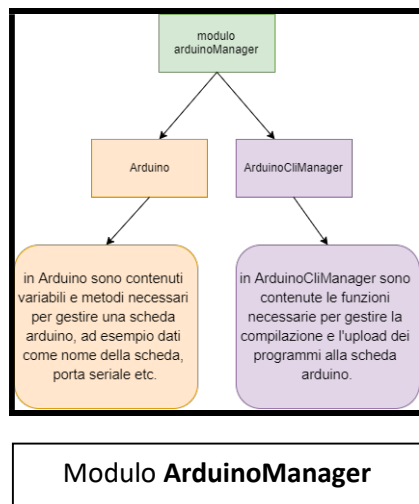
Il programma è stato realizzato utilizzando il linguaggio di programmazione Java con gli obiettivi di garantire una migliore portabilità e la funzionalità multiplatforma.

Il codice del progetto è suddiviso in 4 moduli, ciascuno dei quali si riferisce alle varie classi necessarie al funzionamento del programma; questa scelta consente di apportare modifiche e aggiornamenti a determinate sezioni del programma, ad esempio per aggiungere delle funzioni rese necessarie in un secondo tempo, senza modificare totalmente il programma originale.

L'applicazione SW comprende i seguenti moduli:

- I. **ArduinoManager**: che gestisce le schede arduino collegate al PC
- II. **GUI**: gestisce le finestre grafiche del programma
- III. **SerialManager**: gestisce le porte seriali connesse al PC
- IV. **Sketch**: gestisce i programmi di arduino (sketch)

### I. Modulo ArduinoManager



Come si evince dall'illustrazione il modulo è diviso in due parti (classi) principali, che vanno a gestire i due diversi ambiti.

ArduinoCliManager la comunicazione è resa possibile grazie al programma `arduino-cli` (<https://github.com/arduino/arduino-cli>) che permette la compilazione e l'upload a delle schede tramite linea di comando.

## II. Modulo GUI

Come già accennato, in questo modulo viene gestito il comparto grafico dell'applicativo.

Il risultato è stato ottenuto utilizzando il *framework* appartenente alle *java foundation classes* (cioè l'insieme delle classi base di java). In realtà, i sottomoduli rappresentati dai rettangoli sono composti da due parti: una classe Java, che gestisce la parte logica e l'apertura e chiusura di eventuali sottomenu e poi un file per ogni finestra con estensione *.form*, che fornisce la rappresentazione grafica della finestra. Nella fase di progettazione si è deciso di sviluppare una struttura con questa tipologia per rendere molto più facile e veloce l'aggiunta in un secondo momento di ulteriori schermate per gestire altre esperienze.

All'apertura del programma verrà visualizzata la finestra MenuPage, che contiene i diversi pulsanti e che poi andrà ad aprire le altre sottofinestre corrispondenti alle singole esperienze.

## III. Modulo SerialManager

Il modulo SerialManager è composto da un solo sottomodulo: la classe SerialManager. Questa classe, attraverso l'uso della libreria [jSerialComm](#), permette di gestire una lista di porte seriali connesse. Si rivela molto utile per semplificare il caricamento da parte dell'utente e può essere anche successivamente ampliata per gestire flussi di dati seriali in input e output.

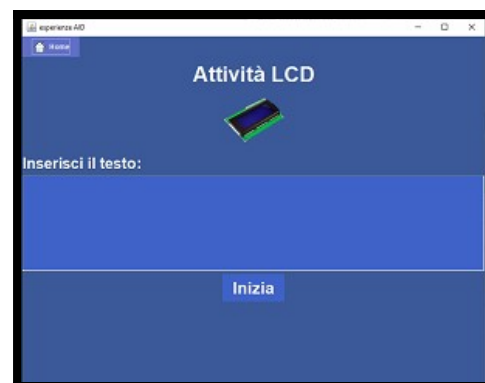
## IV. Modulo sketch

Anche il modulo sketch è composto da una sola classe e gestisce i vari sketch da caricare sulla scheda. E' in grado di gestire la modifica dei dati che dovranno rimanere costanti durante tutta la durata dell'esecuzione dello sketch su Arduino (ad esempio le porte da impostare in input e output).

## V. Pagine finali sul touchscreen



Pagina del menù principale sul touchscreen con le scelte da eseguire



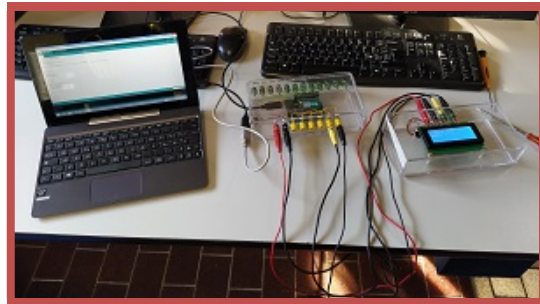
Pagina del menù LCD con il messaggio da inserire da parte dello studente

#### 4) Conclusioni - Collaudo

Attraverso il collaudo ci siamo proposti, oltre ad un controllo e un *debug* funzionale di tutto il sistema, di verificare le ipotesi sperimentali e il metodo applicato direttamente con gli studenti coinvolti. Una volta effettuato l'upload del software lo studente è pronto per eseguire il cablaggio tra il box di Arduino e quello del componente scelto. Il sistema provvede a seguire lo studente con una guida interattiva generata dall'applicazione sullo schermo che lo guida al cablaggio di ciascuna connessione elettrica. Quando tutti i pin sono collegati si chiede allo studente di verificare il risultato dell'esperienza che ha realizzato.



**Menu di scelta dell'esperienza sul nostro touchscreen ASUS Transformer**



**Progetto concluso!  
Collaudo touchscreen+box Arduino+box Display LCD**

I vantaggi che sono emersi al termine del collaudo sono i seguenti:

- ❖ Semplicità di utilizzo del cablaggio rispetto a quello tradizionale con conseguente riduzione delle possibilità di errore anche grazie al supporto del tutorial grafico
- ❖ Possibilità di svolgere l'esperienza nell'arco delle 2 ore di laboratorio canoniche, senza suddividerla in lezioni successive
- ❖ Maggiore concentrazione sugli aspetti tecnici, grazie alla metodologia didattica alternativa
- ❖ Potenziamento dell'autostima e della gratificazione degli studenti in quanto riescono a condurre l'esperienza in modo quasi autonomo senza la presenza dell'insegnante di sostegno e riescono a concludere nei tempi prestabiliti per la classe

## 5) Conclusioni - Miglioramenti, nuove alternative e possibili utilizzi

- Possibilità di interazione audio con microfono/sintetizzatore (per eventuali difficoltà nella scrittura)
- Possibilità di rendere modulare tutte le applicazioni
- Modulo SW con la funzione di controllo automatico delle connessioni effettuate
- Convertire l'attuale cablaggio elettrico con quello magnetico (per migliorare la sicurezza e la facilità dei collegamenti)
- Modifica della parte SW in modo da adattare il livello dello sviluppo del programma in base alle reali esigenze dello studente
- Ingegnerizzazione industriale della parte hardware con box professionali in modo da rendere possibile anche l'intercambiabilità tra dispositivi con la medesima piedinatura tra *shields*, trasduttori, attuatori. Tale soluzione sarebbe anche di grande aiuto durante le dimostrazioni nel corso degli Open Day e ministage per l'orientamento in ingresso degli studenti delle Secondarie di Primo grado, momento in cui non vi è disponibilità di tempo per programmare ed assemblare esperienze con Arduino
- Estendere la soluzione ad altre scuole e, attraverso il contributo di altri studenti e docenti, migliorare ulteriormente il prototipo
- Una analisi di mercato ci ha permesso di rilevare che, al momento, non è disponibile in commercio un sistema integrato di questo tipo e quindi stiamo già pensando anche ad una forma di brevetto per l'intero sistema.

## 6) Conclusioni - Ringraziamenti

La realizzazione di questo progetto ci ha fornito la possibilità di applicare le nostre conoscenze nelle discipline tecniche di Elettronica ed Informatica in modo propositivo e veramente utile all'interno del nostro Istituto su di una problematica, a volte, molto delicata. Pertanto vogliamo ringraziare in modo sentito i docenti del nostro Consiglio di Classe che ci hanno consentito di lavorare al progetto anche durante l'orario curricolare ed il Personale tutto della Scuola che ha reso possibile il nostro lavoro.

Hamza Haddou

Lorenzo Lucca