

Sistema didattico per Arduino

Titolo del progetto: Sistema didattico per Arduino
Alunno/a: Paolo Weishaupt, Carlo Pezzotti
Classe: Info I3AC
Anno scolastico: 2018/2019
Docente responsabile: Adriano Barchi, Luca Muggiasca, Francesco Mussi, Elisa Nannini

Indice

1	Introduzione	3
1.1	Informazioni sul progetto	3
1.2	Abstract	3
1.3	Scopo	3
	Analisi	3
1.4	Analisi del dominio	3
1.5	Analisi e specifica dei requisiti	3
1.6	Analisi dei costi	4
1.7	Pianificazione	1
1.8	Analisi dei mezzi	1
1.8.1	Software	1
1.8.2	Drivers	1
1.8.3	Hardware	1
1.8.4	Specifiche Hardware	2
2	Progettazione	2
2.1	Design dell'architettura del sistema	2
2.2	Design procedurale	3
3	Implementazione	5
4	Test	5
4.1	Protocollo di test	6
4.2	Risultati test	8
4.3	Mancanze/limitazioni conosciute	8
5	Consuntivo	8
6	Conclusioni	8
6.1	Sviluppi futuri	9
6.2	Considerazioni personali	9
7	Bibliografia	9
7.1	Bibliografia per articoli di riviste	9
7.2	Bibliografia per libri	9
7.3	Sitografia	9
8	Allegati	9

Indice delle figure

Figura 1:	Gantt preventivo	1
Figura 2:	Schema elettrico	2
Figura 3:	Circuito elettrico	2

1 Introduzione

1.1 Informazioni sul progetto

Allievi coinvolti: Paolo Claudio Weishaupt, Carlo Pezzotti

Classe: Informatica 3AC presso la Scuola di Arti e Mestieri a Trevano

Docenti responsabili: Adriano Barchi, Luca Muggiasca, Francesco Mussi, Elisa Nannini

Data inizio: 14 / 11 / 2018

Data fine: 25 / 01 / 2019

1.2 Abstract

In this project you will find some very useful libraries to begin your training with the Arduino platform and a very well made manual on how to use them. At the end of the manual you will be able to use the very basic components of the Arduino and you will know the basics of its language.

1.3 Scopo

Lo scopo del progetto è quello di fornire delle librerie utili alla dimostrazione di ciò che viene spiegato nella sezione Informatica della SAMT riguardanti l'ambito Arduino. Il tutto verrà accompagnato da un dettagliato manuale d'uso che spiegherà come collegare i vari moduli al DigiSpark e come usare i metodi contenuti.

Analisi

1.4 Analisi del dominio

Il cliente vuole una libreria che permetta di chiamare delle funzioni che facilitino la stesura del codice. L'idea sarebbe quella di semplificare il più possibile il codice che dovrà utilizzare un terzo utente. L'utente finale saranno dei ragazzi di terza media, quindi con competenze informatiche basse o addirittura nulle.

Le librerie dovranno utilizzare tutti gli attuatori utilizzabili sul DigiSpark. Il mio team ha inizialmente il compito di sviluppare una libreria su un led RGB e tre bottoni.

1.5 Analisi e specifica dei requisiti

ID	REQ-001
Nome	Digispark
Priorità	1
Versione	1.0
Nota	Digispark è un componente elettronico che serve per mettere insieme programmazione e elettronica.
Sub-ID	Requisito
001	Digispark deve funzionare completamente.

ID	REQ-002
Nome	Attuatori
Priorità	1
Versione	1.0
Nota	Per far implementare dell'elettronica con la programmazione bisogna avere dei componenti elettronici chiamati attuatori.
Sub-ID	Requisito
001	Attuatori deve funzionare completamente.

ID	REQ-003
Nome	Ambiente di programmazione Arduino
Priorità	1
Versione	1.0
Nota	Per programmare si può utilizzare un qualsiasi editore di testo, per caricare il codice sulla scheda bisogna però utilizzare l'IDE(Integrated development environment) di Arduino.
Sub-ID	Requisito
001	Computer funzionante.

ID	REQ-004
Nome	Funzione che legge stato bottone
Priorità	2
Versione	1.0
Nota	Per realizzare del codice facile bisogna scrivere una funzione che in base al pin passato come argomento riconosce e ritorna lo stato di un bottone, se è premuto oppure no.
Sub-ID	Requisito
001	Bottoni funzionanti collegati ad una breadboard o veroboard.

ID	REQ-005
Nome	Funzione che incrementa valore led
Priorità	2
Versione	1.0
Nota	Ci dovrà essere una funzione nascosta all'utente, o che comunque non potrà utilizzare, che incrementerà la potenza di uscita dei pin collegati al Led RGB. Questi valori dovranno essere 3: R, G, B e dovranno avere un massimo di potenza 255 e un minimo di 0.
Sub-ID	Requisito
001	Funzione lettura bottone.

ID	REQ-006
Nome	Funzione che scrive il valore
Priorità	2
Versione	1.0
Nota	Per realizzare del codice facile bisogna scrivere una funzione che in base al pin passato come argomento scrive lo stato che può essere analogico oppure digitale. Nel nostro caso dove bisogna utilizzare un Led RGB dovremmo utilizzare delle uscite analogiche per rendere migliore l'esperienza.
Sub-ID	Requisito
001	Led funzionante collegato ad una breadboard o veroboard.

1.6 Analisi dei costi

Nome	Salario	Ore di lavoro	Costo
Paolo Weishaupt	62 CHF/h	54	3348
Carlo Pezzotti	62 CHF/h	54	3348

In base a quanto calcolato sopra, con un salario di 62 CHF all'ora ognuno di noi dovrebbe costare 3348 CHF all'azienda.

1.7 Pianificazione

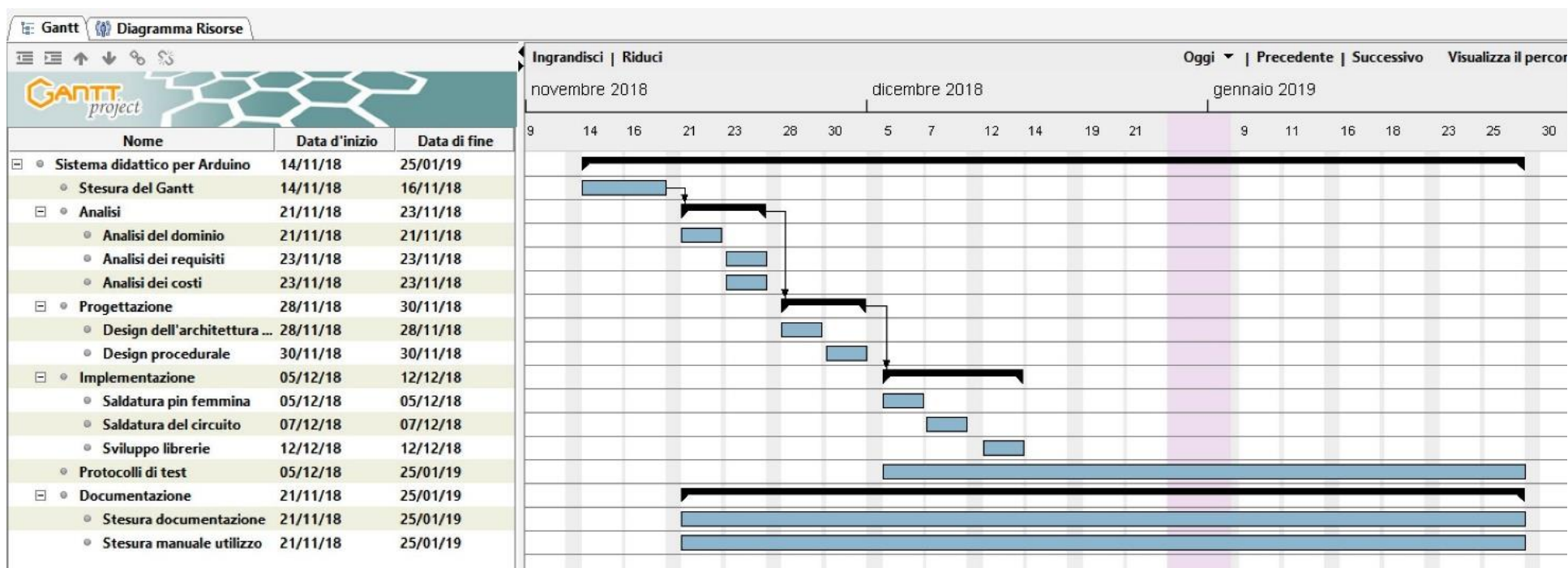


Figura 1: Gantt preventivo

Titolo del progetto: Sistema didattico per Arduino
Alunno/a: Paolo Weishaupt, Carlo Pezzotti
Classe: Info I3AC
Anno scolastico: 2018/2019
Docente responsabile: Adriano Barchi, Luca Muggiasca, Francesco Mussi, Elisa Nannini



Scuola Arti e Mestieri Trevano

Sezione informatica

1.8 Analisi dei mezzi

1.8.1 Software

Nome software	Versione
Arduino IDE	1.8.7
Fritzing	0.9.3b
Atom	1.33.0
GanttProject	2.8.9 B2335
Microsoft Office Professional Plus	2016 16.0.4266.1001
GitHub Desktop	1.5.1
OneDrive(solo Paolo)	18.222.1104.0007

1.8.2 Drivers

Nome driver	Versione
Digistump LLC (digistump.com)	09/02/2014 5.1.2600.1
Digistump LLC	08/16/2014 1.1.0.0
Libusb-win32 Digispark Bootloader	01/17/2012 1.2.6.0
Libusb-win32 DigiUSB	09/02/2014 1.2.6.1

1.8.3 Hardware

Il progetto verrà sviluppato su una scheda DigiSpark facilmente reperibile online. La scheda ha una limitazione ossia il suo carente numero di porte che ci limita all'utilizzo di pochi attuatori.

Carlo userà un HP Omen del 2016 con Windows 10 Pro mentre io utilizzerò un Huawei Matebook X Pro del 2018 con Windows 10 Home.

Titolo del progetto: Sistema didattico per Arduino
Alunno/a: Paolo Weishaupt, Carlo Pezzotti
Classe: Info I3AC
Anno scolastico: 2018/2019
Docente responsabile: Adriano Barchi, Luca Muggiasca, Francesco Mussi, Elisa Nannini

1.8.4 Specifiche Hardware

HP Omen 2016

Processore	Intel Core i7-6700HQ
Scheda grafica	NVIDIA GeForce GTX 965M
Ram	16GB
Display	15.6" 1920 x 1080px
Storage	128GB SSD + 1TB HDD

Huawei MateBook X Pro

Processore	Intel Core i7-8550U
Scheda grafica	NVIDIA GeForce GTX 150M
Ram	8 GB
Display	13.9" 3000x 2000px
Storage	512GB SSD

DigiSpark

- Supporto per Arduino IDE 1.0+ (OSX / Win / Linux)
- Alimentazione tramite USB o sorgente esterna - 5v o 7-35v (consigliato 12v o meno, selezione automatica)
- Regolatore con uscita 5V - 500mA
- Integrato USB incorporato
- 6 pin I / O (2 usano solo USB se il programma comunica attivamente tramite USB, altrimenti è possibile utilizzare tutti i 6 pin anche se si sta programmando tramite USB)
- 8k Memoria Flash (circa 6k dopo il bootloader)
- I2C e SPI
- PWM su 3 piedini (altri con Software PWM)
- ingresso ADC su 4 pin
- LED di alimentazione e LED di prova/stato

2 Progettazione

2.1 Design dell'architettura del sistema

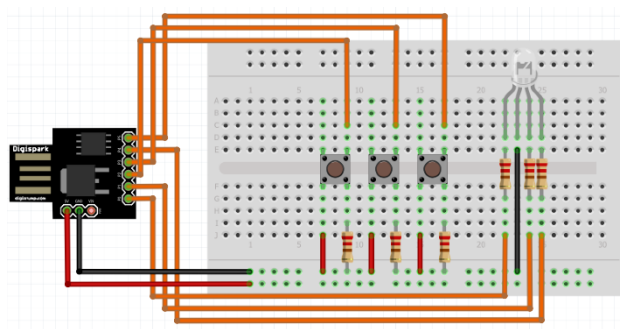


Figura 2: Schema elettrico

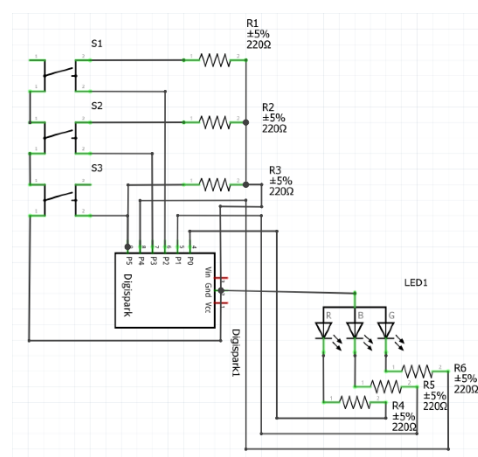
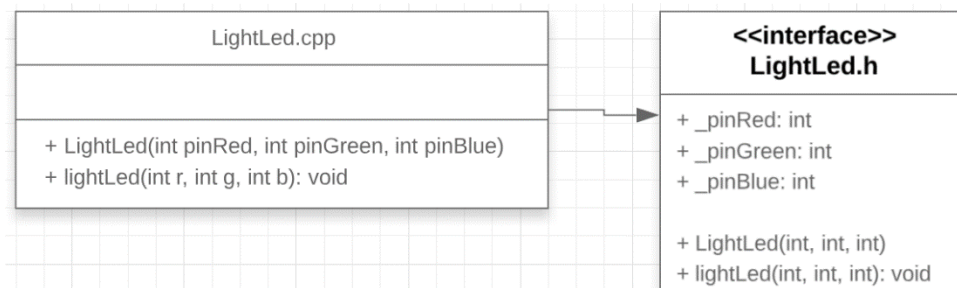


Figura 3: Circuito elettrico

2.2 Design procedurale

In tutte le librerie ci sarà un import della libreria Arduino.h per importare i metodi di base dell'Arduino.

Libreria per un Led RGB:



Classi:

- LightLed.cpp
- LightLed.h

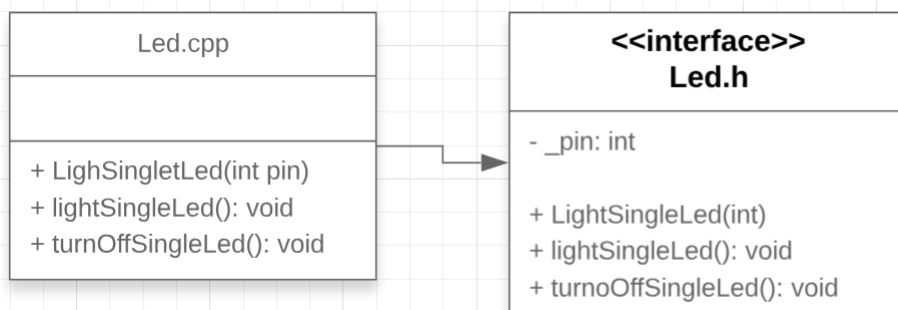
Attributi:

- int _pinRed: pin collegato al colore rosso
- int _pinGreen: pin collegato al colore verde
- int _pinBlue: pin collegato al colore blu

Metodi:

- LightLed (int pinRed, int pinGreen, int pinBlue): costruttore che dati i tre pin del Led RGB istanzia un oggetto di tipo LightLed
- void lightLed (int r, int g, int b): metodo che accende i led del Led RGB tutti assieme in base al valore passato come parametro

Libreria per un Led:



Classi:

- Led.cpp
- Led.h

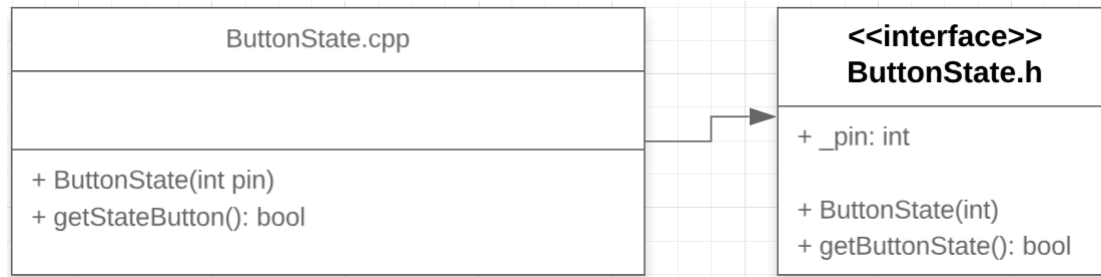
Attributi:

- int _pin: pin del Led

Metodi:

- LightSingleLed (int pin): costruttore che dato il pin del Led istanzia un oggetto di tipo LightSingleLed
- void lightSingleLed (): metodo che accende il Led
- void turnOffSingleLed (): metodo che spegne il Led

Libreria per un Bottone:



Classi:

- ButtonState.cpp
- ButtonState.h

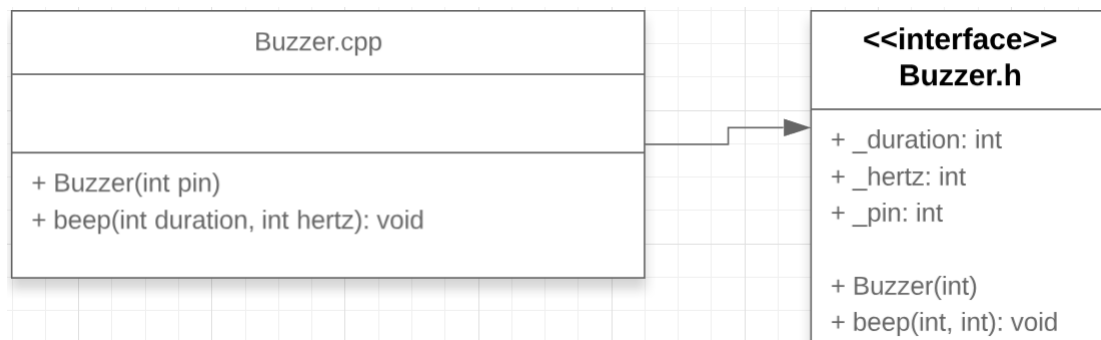
Attributi:

- int _pin: pin del bottone

Metodi:

- ButtonState (int pin): costruttore che dato il pin del bottone istanzia un oggetto di tipo Button
- bool getStateButton (): metodo che ritorna lo stato del bottone. Se è premuto ritorna 1 invece se è spento ritorna 0

Libreria per un Piezo Buzzer:



Classi:

- Buzzer.cpp
- Buzzer.h

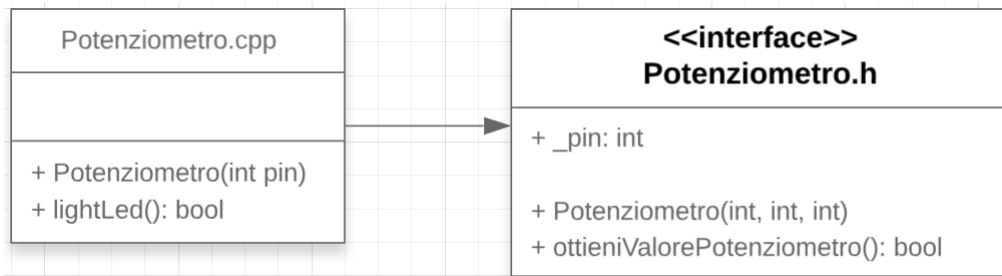
Attributi:

- int _duration: durata del suono da emettere
- int _hertz: frequenza del suono da emettere
- int _pin: pin del buzzer

Metodi:

- Buzzer (int pin): costruttore che dato il pin del buzzer istanzia un oggetto di tipo Buzzer
- void beep (int duration, int hertz): metodo che data una certa frequenza e una durata, fa suonare il piezo buzzer.

Libreria per un Potenzenziometro:



Classi:

- Potenziometro.cpp
- Potenziometro.h

Attributo:

- int _pin: pin del potenziometro

Metodi:

- Potenziometro (int pin): costruttore che dato il pin del buzzer istanzia un oggetto di tipo Potenziometro
- void ottieniValorePotenziometro(): Metodo che ritorna il valore del potenziometro

3 Implementazione

3.1 Libreria per un Led RGB

3.1.1 LightLed.h

Questa parte di codice serve per impedire che venga importato lo stesso file header più di una volta altrimenti verrebbero generati degli errori a causa della presenza di più metodi definiti allo stesso modo.

```

8  #include "LightLed.h"
9  #import <Arduino.h>
    
```

Così si definiscono gli attributi e i metodi che devono essere implementati dalle classi che utilizzano questa interfaccia.

```

11  // Library interface description
12  class LightLed
13  {
14      public:
15          int _pinRed;
16          int _pinGreen;
17          int _pinBlue;
18          LightLed(int, int, int);
19          void lightLed(int, int, int);
20  };
    
```

3.1.2 LightLed.cpp

Grazie a queste due righe di codice includo l'header che definisce gli attributi e i metodi da utilizzare e importo il file Arduino.h che definisce i metodi di base di Arduino.

```
8  #include "LightLed.h"
9  #import <Arduino.h>
```

Il costruttore ci permette di definire i tre pin passati come argomento in output e di assegnarne il loro valore agli attributi corrispondenti.

```
18  LightLed::LightLed(int pinRed, int pinGreen, int pinBlue)
19  {
20      pinMode(pinRed,OUTPUT);
21      pinMode(pinGreen,OUTPUT);
22      pinMode(pinBlue,OUTPUT);
23      _pinRed = pinRed;
24      _pinGreen = pinGreen;
25      _pinBlue = pinBlue;
26  }
```

Il metodo lightLed riceve come parametri i valori da settare ai pin in output per accendere il Led RGB. Al metodo AnalogWrite bisogna passare il pin da accendere e il suo valore.

```
35  void LightLed::lightLed(int r, int g, int b){
36      analogWrite(_pinGreen, g);
37      analogWrite(_pinRed, r);
38      analogWrite(_pinBlue, b);
39  }
```

4 Test

4.1 Protocollo di test

Test Case:	TC-001	Nome:	Digispark
Riferimento:	REQ-001		
Descrizione:	Per il corretto funzionamento bisogna verificare che il digispark sia correttamente funzionante		
Prerequisiti:	Nulla.		
Procedura:	1. Collegare il digispark al computer e verificare che il led rosso si accenda		

	2. Per verificare se tutti i pin funzionano correttamente bisogna fare un piccolo programma dove sequenzialmente si fanno accendere i led.
Risultati attesi:	Se tutto dovesse andare a buon fine allora mi aspetterai che il digispark e tutti pin funzionino correttamente.

Test Case:	TC-002	Nome:	Attuatori
Riferimento:	REQ-002		
Descrizione:	Per il giusto funzionamento di un attuatore bisogna verificare la scheda tecnica di esso quindi come utilizzarlo.		
Prerequisiti:	Una scheda digispark funzionante vedi TC-001		
Procedura:	<ol style="list-style-type: none"> 1. Recarsi sulla scheda tecnica dell'attuatore verificato e visualizzare il giusto funzionamento (es. I poli) 2. Provare a fare un piccolo circuito o utilizzare uno già esistente e verificare che seguendo le istruzioni scritte sulla scheda tecnica l'attuatore svolga il lavoro desiderato. 		
Risultati attesi:	Se tutto dovesse andare a buon fine allora il risultato finale sarà uguale a quello aspettato (es. un led si accende se lo collego)		

Test Case:	TC-003	Nome:	Ambiente ide di arduino
Riferimento:	REQ-003		
Descrizione:	Per poter programmare con Arduino/digispark bisogna avere un ambiente di sviluppo e di carica dati. Arduino ci fornisce il suo.		
Prerequisiti:	Computer funzionante.		
Procedura:	<ol style="list-style-type: none"> 1. Scaricare dal sito ufficiale l'IDE di Arduino 2. Provare a lanciare il programma 3. Far partire un codice di esempio e verificare che il digispark faccia quello che deve fare 		
Risultati attesi:	Se tutto dovesse andare una volta che si esegue un codice allora digispark farà quello per cui è stato programmato (es. codice di prova blinking led -> il digispark farà lampeggiare il led di cui è fornito)		

Test Case:	TC-004	Nome:	Funzione che legge stato bottone
Riferimento:	REQ-004		
Descrizione:	Se il programma e il circuito fornito dovessero funzionare allora con l'avvio del programma digispark sarà capace di leggere lo stato di un bottone con una semplice funzione		
Prerequisiti:	Tutti i requisiti precedenti.		
Procedura:	<ol style="list-style-type: none"> 1. Istanziare un nuovo oggetto ButtonState (vedi manuale allegato) 2. Chiamare la funzione e verificare che l'Arduino legga ciò che deve leggere 		

Risultati attesi:	Se il programma è stato scritto in modo giusto allora l'Arduino svolgerà l'operazione per il quale è stato programmato.
--------------------------	---

Test Case:	TC-005	Nome:	Funzione che incrementa stato del led
Riferimento:	REQ-005		
Descrizione:	In base alla nostra logica, quando un bottone viene premuto il led RGB incrementa il colore (in base al bottone).		
Prerequisiti:	Funzione di lettura bottone funzionante.		
Procedura:	<ol style="list-style-type: none"> 1. Istanziare un nuovo oggetto ButtonState (vedi manuale allegato) 2. Quando il bottone viene premuto incrementare il valore del colore rosso/verde/o blu 		
Risultati attesi:	Se si preme il bottone si incrementa il colore.		

Test Case:	TC-006	Nome:	Funzione che scrive il valore desiderato
Riferimento:	REQ-006		
Descrizione:	Una volta che si ha il colore bisogna solo scrivere il colore sul led RGB, ciò si fare grazie alla libreria da noi ideata e scritta (vedi manuale)		
Prerequisiti:	Tutti i requisiti precedenti		
Procedura:	<ol style="list-style-type: none"> 1. Utilizzare la funzione da noi ideata e scritta per scrivere i dati sul led RGB, (vedi manuale) 		
Risultati attesi:	Quando si preme un bottone il valore viene incrementato e scritto.		

4.2 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.

4.3 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.

5 Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

6 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.

6.1 Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

6.2 Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

7 Bibliografia

7.1 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,

7.2 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.

7.3 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.

8 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
- ...