Sistema didattico per Arduino

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 6

1.8 Analisi dei mezzi 6

1.8.1 Software 6

1.8.2 Hardware 6

2 Progettazione 7

2.1 Design dell’architettura del sistema 7

2.2 Design dei dati e database 7

2.3 Design delle interfacce 7

2.4 Design procedurale 7

3 Implementazione 8

4 Test 8

4.1 Protocollo di test 8

4.2 Risultati test 9

4.3 Mancanze/limitazioni conosciute 9

5 Consuntivo 9

6 Conclusioni 9

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 10

# Introduzione

## 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

## Abstract

*In this project you will find some very useful libraries to begin you 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 basics components of the Arduino and you will know the basics of it’s language.*

## 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

## 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.

## 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. |

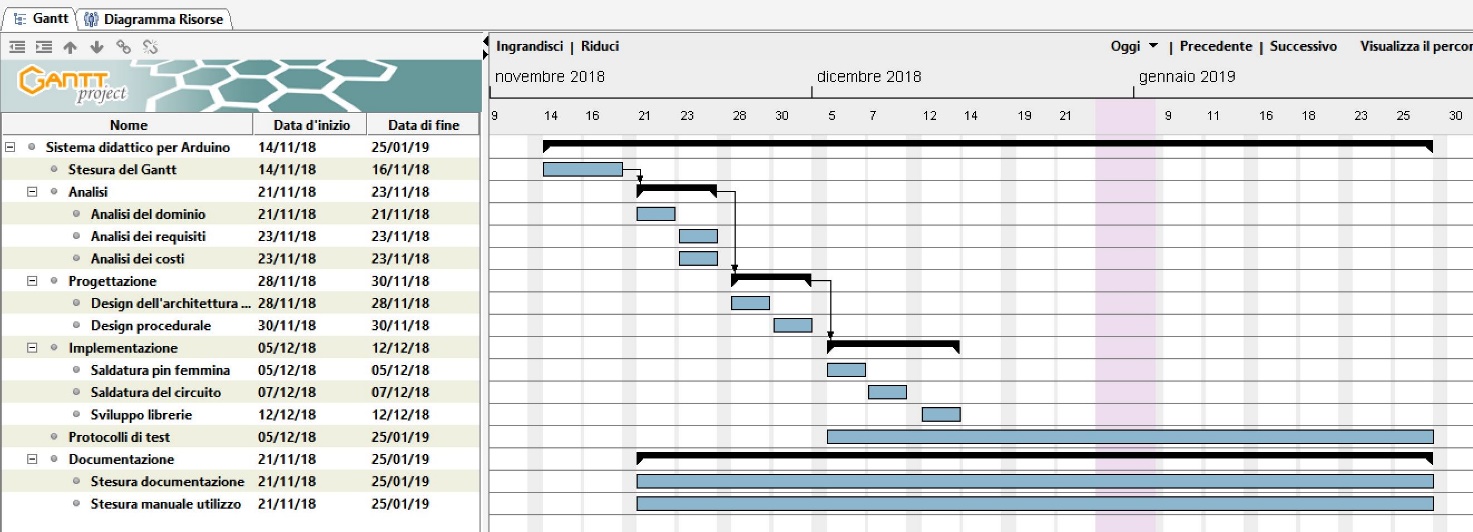
|  |  |
| --- | --- |
| ID | REQ-005 |
| Nome | Funzione che incrementa valore led |
| Priorità | 2 |
| Versione | 1.0 |
| Nota | Ci dovrà essere una funzione nascosta all’utente, o che comunque non potrà utilizzare, che incrementerà la potenza di uscita dei pin collegati al Led RGB. Questi valori dovranno 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. |

## Analisi dei costi

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

## Pianificazione



## Analisi dei mezzi

### Software

Arduino IDE 1.8.7

Fritzing 0.9.3b

Atom 1.33.0

GanttProject 2.8.9 B2335

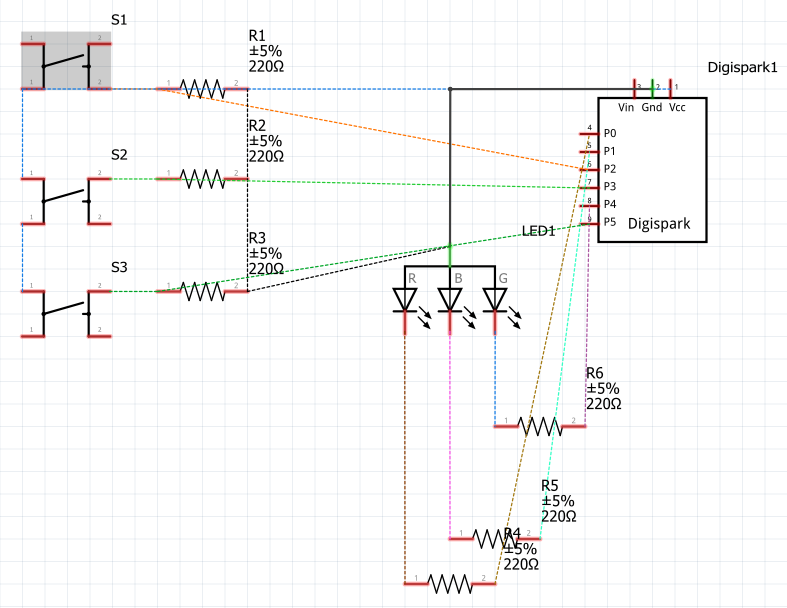
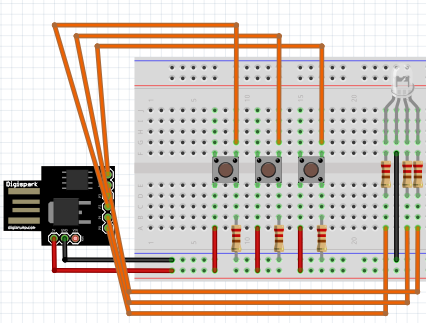
### Hardware

Il progetto verrà sviluppato su una scheda DigiSpark facilmente reperibile online. La scheda ha una limitazione ossia il suo carente numero di porta 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.

# Progettazione

## Design dell’architettura del sistema



## Design delle interfacce

Descrizione delle interfacce interne ed esterne del sistema e dell’interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# 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

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## 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

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

# 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

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## 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
* …