Sistema didattico per Arduino con libreria per attuatori e relativa documentazione

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

**Titolo**: Sistema didattico per Arduino con libreria per attuatori e relativa documentazione

**Allievi**: Mattia Lazzaroni, impiegato nello svolgimento del progetto.

Mattia Toscanelli, impiegato nello svolgimento del progetto.

**Classe**: I3AC

**Docenti**: Adriano Barchi, Luca Muggiasca, Francesco Mussi, Massimo Sartori

**Sezione scuola**: Scuola arti e mestieri Trevano

**Materia**: Modulo 306 – Progetti

**Data inizio**: 14.11.2018

**Fine**: 08.02.2019

## Abstract

E’ una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

* **Background/Situazione iniziale**
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

*As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing*

## Scopo

## Lo scopo del progetto è quello di realizzare un prodotto didattico dedicato agli allievi di terza media che arriveranno nelle giornate di porte aperte Promtec. Il nostro progetto consentirà all’utente di ambientarsi con la piattaforma elettronica Arduino (nel nostro caso mini DigiSpark) grazie a delle guide semplici ed intuitive che spiegheranno come utilizzarlo e come programmarlo. Per far capire al meglio il concetto di programmazione il prodotto offre delle librerie create da noi contenenti dei metodi a cui sono stati attribuiti nomi semplici ed intuitivi. Al nostro gruppo sono stati assegnati i seguenti componenti: - Potenziometro - ServoMotore - Ultrasuoni

## Analisi

## Analisi del dominio

Bisogna trovare un metodo che aiuti i ragazzi di terza media, giunti alla scuola arti e mestieri Trevano durante le porte aperte Promtec, ad approciarsi al mondo della programmazione. All’inzio della giornata lo studente riceverà un DigiSpark (un Arduindo molto piccolo, ma che svolge le stesse funzioni). L’obbiettivo è quello di creare delle librerie per quest’utlimo contenteti metodi che semplificano la stesura del codice. Tutto ciò viene viene accompagnato con dei manuali utente in cui sono scritte dettagliate descrizioni su come funziona la liberiria e su come devono venir utilizzati i vari metodi con tanto di esempi. Alla fine della giornata Il ragazzo tornarà a casa con il DigiSpark, un piccolo manuale d’uso e sicuramente con delle conoscenze in più per quanto riguarda la programmazione.

## Analisi e specifica dei requisiti

Il committente deve ricevere un DigiSpark e un guida per capirne il funzionamento. Il prodotto deve svolgere determinate azioni, come far accendere un led tramite un potenziometro. Queste funzioni sono implementate tramite il software Arduino 1.8.7. L’utente dovrebbe consultare la guida per capire come montare il circuito e come usare il proprio Arduino e le librerie presenti in esso da noi implementate. L’allievo consultando la guida dovrà capire tutto ciò che dovrà fare tramite una spiegazione semplice ed efficace, anche se egli non si è mai applicato prima nell’elettronica. Come minimo il prodotto deve far si che tutte le librerie presenti funzionino e raggiungano il risultato atteso al 100%, senza bug o mancanze. Ovviamente il circuito dovrà essere sicuro e senza la presenza di rischi che possano ferire un visitatore. Quando l’utente avrà finito la sua attività avrà la possibilità di tenere il lavoro da lui svolto come ricordo della sua visita.

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Saldamento dei pin DigiSpark |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Si necessita un modulo da 6 pin femmina. |
| **002** | Si necessita di un saldatore. |
| **003** | Si necessita dello stagno. |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Saldamento del circuito |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Si deve avere lo schema del circuito. |
| **002** | Si necessita un LED rosso. |
| **003** | Si necessita una resistenza da 330Ω. |
| **004** | Si necessita di un potenziometro. |
| **005** | Si necessita dello stagno. |
| **006** | Si necessita di una veroboard per poter saldate tutti i componenti. |
| **007** | Si necessita del filo argentato per fare dei ponti. |
| **008** | Si necessita un saldatore. |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Creazione delle librerie |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Per poter implementare le librerie bisogna completare i requisiti REQ-001 e REQ-002. |
| **Sottorequisiti** | |
| **001** | Il codice deve essere ben indentato. |
| **002** | Il codice deve essere commentato. |
| **003** | I nomi dei metodi devono essere di facile compresione da parte dell’utente. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Creazione della guida |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Per poter scrivere la guida bisogna completare il requisito REQ-003. |
| **Sottorequisiti** | |
| **001** | Bisogna installare la scheda Digispark. |
| **002** | Deve essere comprensibile a tutti gli utenti. |
| **003** | Deve contenere la spiegazione di tutte le librerie. |
| **004** | Deve avere degli esempi per aiutare la comprensione dell’utente. |

## Analisi dei costi

|  |  |  |  |
| --- | --- | --- | --- |
| Categoria | Ore di lavoro | Costo all’ora | Costo totale |
| Manodopera Mattia Lazzaroni | 50 h | 62 CHF | 3069 CHF |
| Manodopera Mattia Toscanelli | 50 h | 62 CHF | 3069 CHF |

## Pianificazione

Abbiamo inserito due pietre miliari, una a fine progettazione, cioè quando il progetto inizierà ad essere concreto; l’altra a fine progetto, quando potrà essere consegnato al cliente. La fase di implementazione sarà quella in cui dedicheremo più tempo, in particolare l’attività “Sviluppo librerie”, in quanto necessita di maggiore cura e inoltre per funzionare correttamente richiede di effetturare molti test.

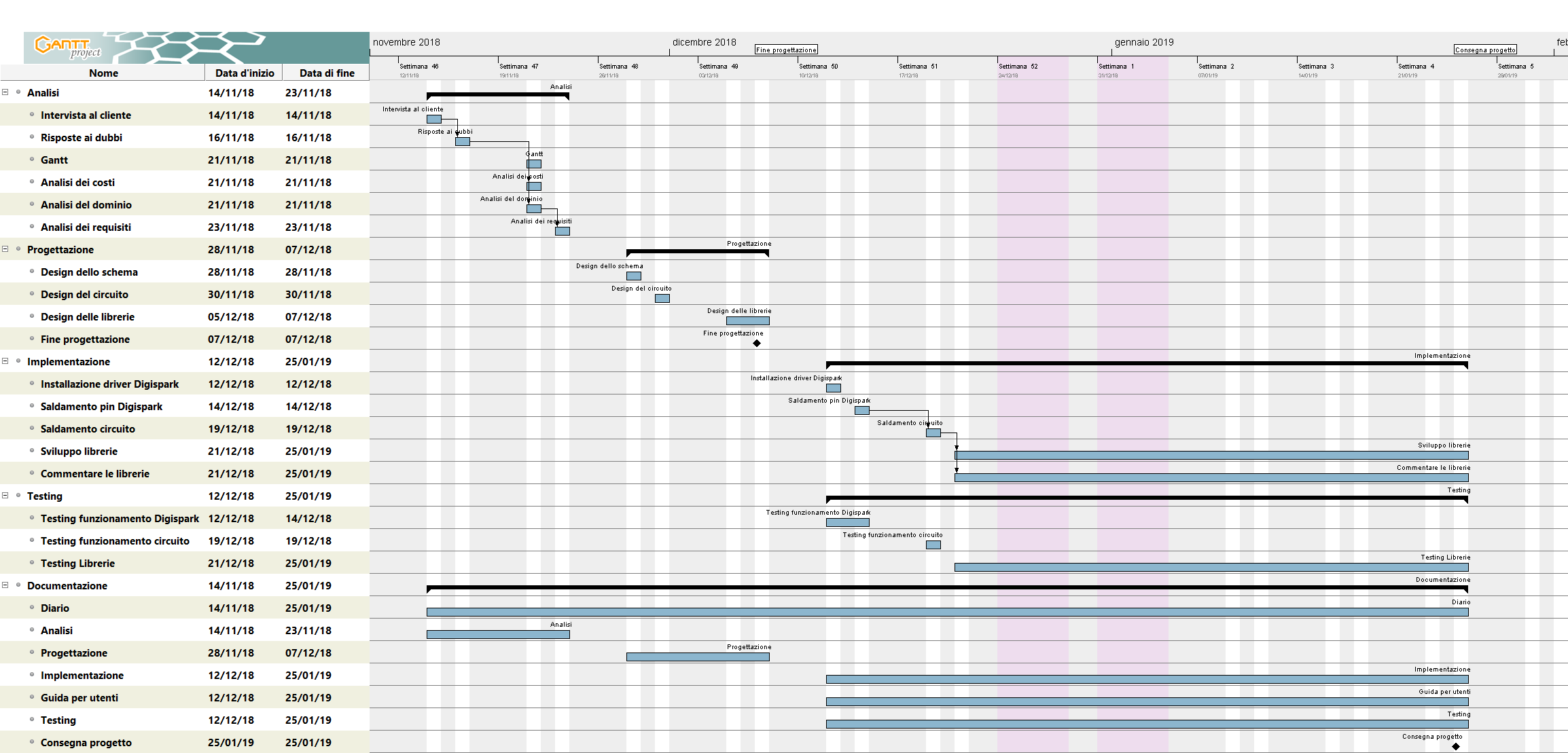


Figura , gantt di pianificazione.

## Analisi dei mezzi

Per la realizzazione di questo progetto si ha bisogno di:

* Un arduino o simile
* Un PC con performance in grado di far girare il software di arduino

### Software

I software utilizzati per la realizzazione di questo progetto sono:

* Word 2016
* Fritzing 0.9.3b
* Arduino 1.8.7
* Power Point 2016
* GanttProject 2.8.9
* Google Chrome

### Hardware

I componenti hardaware utilizzati per la realizzazione del progetto sono:

* Digispark USB Development Board
* Caratteristiche computer Mattia Toscanelli:
  + Modello: Huawei Matebook X Pro
  + Processore: i7 8550u
  + RAM: 8GB LPDDR3 2133 MHz
  + GPU: NVIDIA MX150
  + SSD: 512 GB
* Caratteristiche computer Mattia Lazzaroni:
  + Modello: Acer Aspire E 15
  + Processore: i7 7500u
  + RAM: 16GB LPDDR4 2133 MHz
  + GPU: NVIDIA 940MX
  + SSD: 256 GB

# Progettazione

## Design degli schemi elettrici

### Schema elettrico potenziometro

Il funzionamento di questo circuito è molto semplice. Per farlo si necessita un diodo rosso, una resistenza da 180Ω e un potenziometro da 10kΩ. Lo scopo di questo circuito è il cambiamento di stato del LED grazie alla rotazione del potenziometro. Per effetture questo circuito bisogna collegare il terminale 3 del potenziometro (vedi Figura 2) a un pin analogico del Digispark (in questo caso il P2). Per avere un cambiamento di luminosità del LED bisogna collegare anch’esso a un pin analogico, nel nostro caso al pin P0.

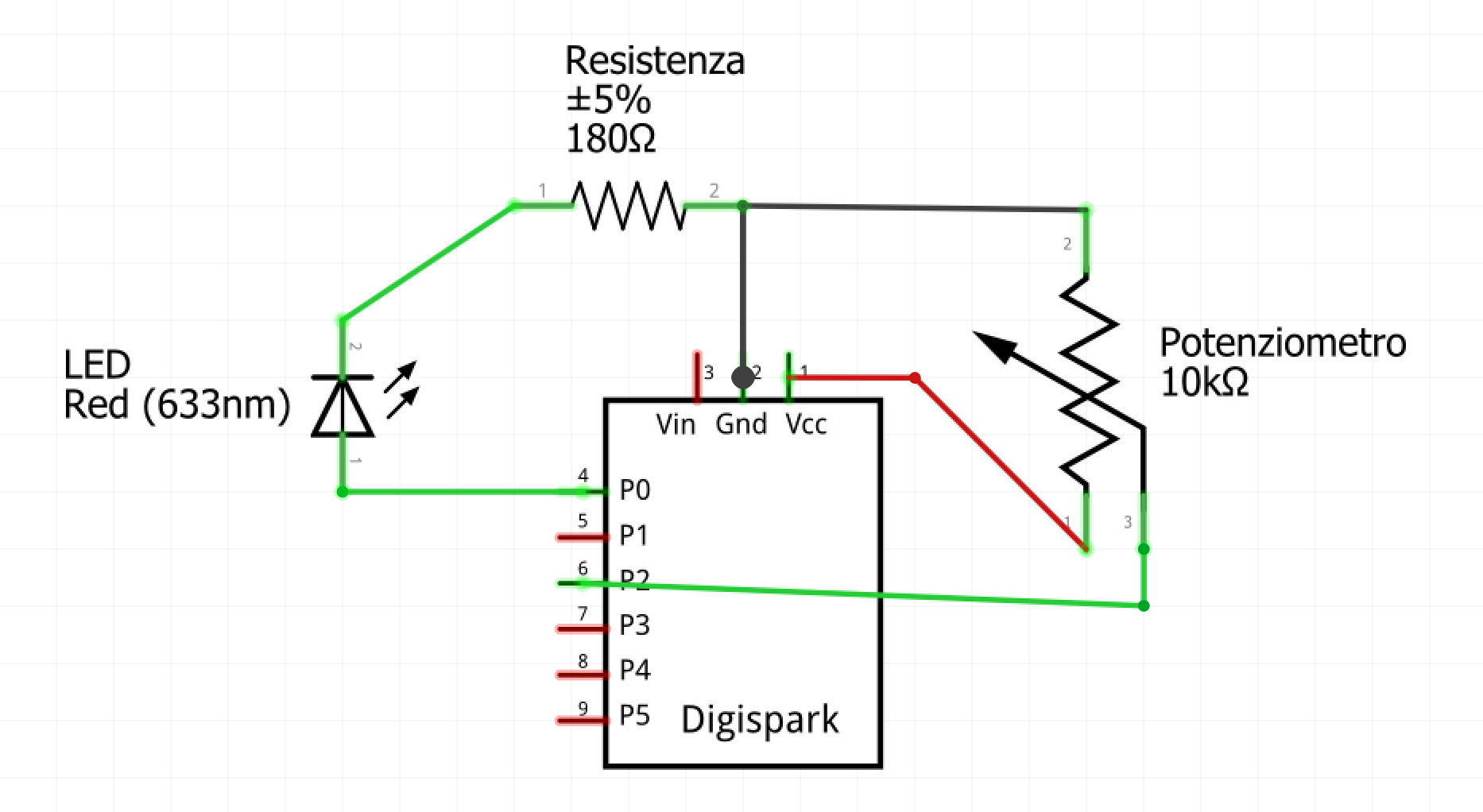


Figura , schema elettrico Potenziometro

### Schema elettrico ServoMotore

Per fare questo circuito si necessita un pulsante e un ServoMotore. Lo scopo di questo circuito è il muovimento del Servomotore grazie alla pressione o meno del pulsante. Per effetture questo circuito bisogna collegare il terminale di pulse del ServoMotore (vedi Figura 3) a un pin analogico del Digispark (in questo caso il pin P3). In seguito bisogna collegare a un pin del Digispark uno dei due terminali 1 del pulsante (vedi Figura 3) per capire quando fa cambiare lo stato del ServoMotore.

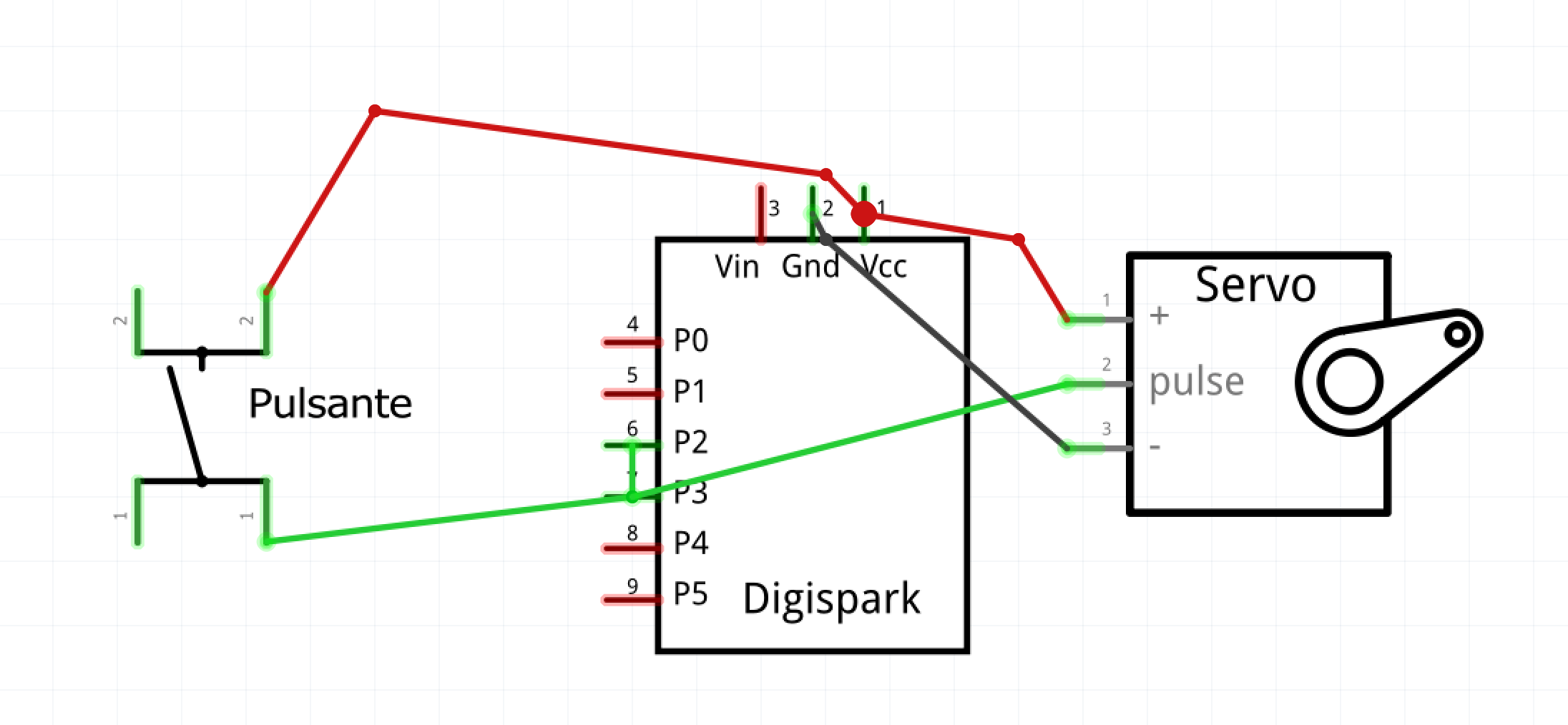


Figure 3, Schema elettrico ServoMotore

### Schema elettirco Ultrasuoni

Per fare questo circuito si necessita un sensore Ultrasuoni e un LED. Lo scopo di questo circuito è il cambiamento di stato del LED in base a quello che legge il sensore Ultrasuoni. Per effetture questo circuito bisogna collegare il terminale di echo dell’Ultrasuoni (vedi Figura 4) a un pin analogico del Digispark (in questo caso il pin P2). In seguito bisogna collegare il pin di trig ad qualsiasi pin del Digispark (in questo caso il pin P3).

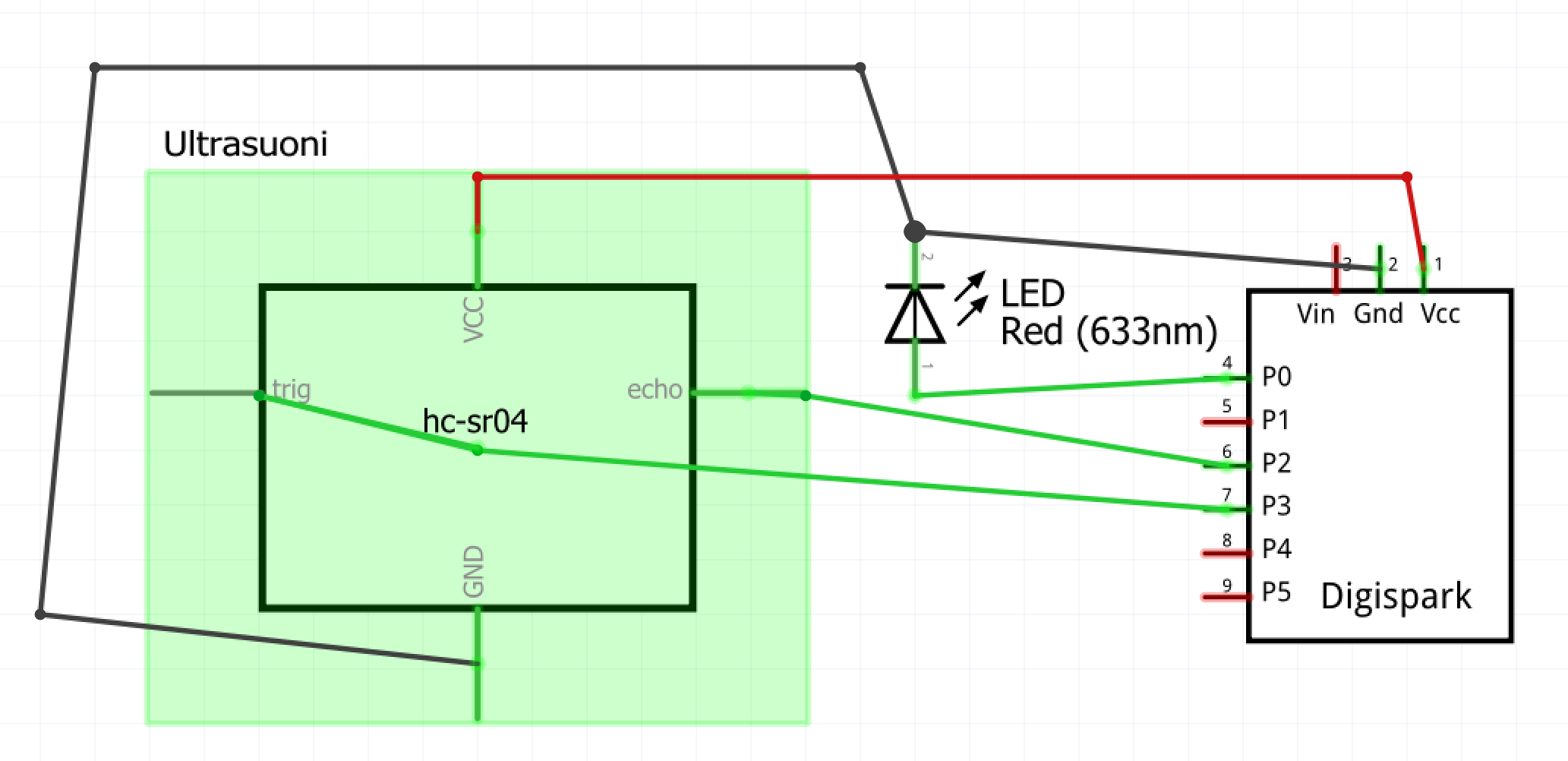


Figure 4, schema elettrico Ultrasuoni

## Design delle librerie

### Libreria Potenziometro

Questa libreria semplica l’utilizzo di un Potenziometro. Contiene un metodo che permette la lettura del Potenziometro in modo rapito e in modo più semplificato.

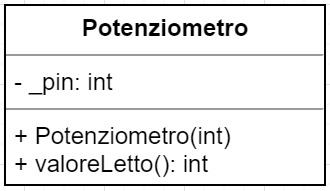


Figura 5, Libreria del potenziometro

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrizione |
| \_pin | Variabile | L’attributo \_pin conterrà il numero del pin a cui sarà collegato il terminale di lettura del potenziometro. |
| Potenziometro(int) | Metodo costruttore | È il metodo costruttore della classe Potenziometro. È composto da un solo paramertro e quest’ultimo corrisponde al terminale di lettura del potenziometro. |
| valoreLetto(): int | Metodo | Questo metodo ritorna il valore letto dal pin analogico in cui è collegato il terminale del potenziometro. Il valore viene letto e covetito in un intervallo che parte da 0 fino a 255. |

### Libreria ServoMotore

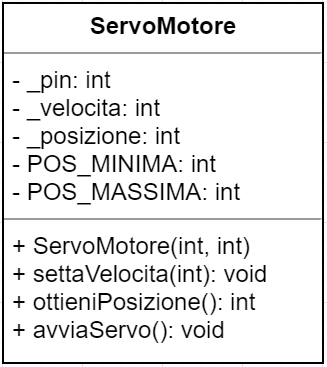
Questa libreria semplica l’ultizzo del ServoMotore. Contiene dei metodi che ottimizzano l’utilizzo del ServoMotore, come il settaggio della velocita e l’ottenimento della posizione.

Figura 6, Libreria ServoMotore

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrizione |
| \_pin | Variabile | L’attributo \_pin conterrà il numero del pin a cui sarà collegato il terminale di lettura del ServoMotore. |
| \_velocita | Variabile | L’attributo \_velocita conterrà la velocita del ServoMotore. Questo valore può variare da -100 a 100. |
| \_posizione | Variabile | L’attributo \_posizione conterrà la posizione dell’elica del ServoMotore. Questo valore parte da 0° fino a 180° |
| POS\_MINIMA | Variabile costante | L’attibuto POSIZIONE\_MIMIMA conterrà la posizione minima dell’elica del ServoMotore (cioè la posizione corrispondente a 0°). |
| POS\_MASSIMA | Variabile costante | L’attibuto POSIZIONE\_MASSIMA conterrà la posizione massima dell’elica del ServoMotore (cioè la posizione corrispondente a 180°). |
| ServoMotore(int,int) | Metodo costruttore | È il metodo costruttore della classe ServoMotore. È composto da due parametri: il primo esprime il pin a cui è collegato il ServoMotore e il secondo esprime la velocità dell’elica. |
| settaVelocita(int) | Metodo | Questo metodo permette si settare la velocità al ServoMotore. All’interno di esso viene controllata la velocità passata come parametro ed eventualemente viene corretta. |
| ottieniPosizione() | Metodo | Questo metodo ritorna la posizione dell’elica del ServoMotore. |
| avviaServo() | Metodo | Questo metodo fa muovere il ServoMotore. |

### Libreria Ultrasuoni

Questa libreria semplifica l’utlizzo dell’Utrasuoni. Contiene un metodo che permettere di sapere velocemente la velocità letta dall’Ultrasuoni.

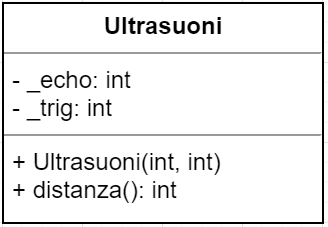


Figura 7, Libreria Ultrasuoni

|  |  |  |
| --- | --- | --- |
| Nome | Tipo | Descrizione |
| \_echo | Variabile | L’attributo \_echo conterrà il numero del pin a cui sarà collegato il terminale di echo. |
| \_trig | Variabile | L’attributo \_echo conterrà il numero del pin a cui sarà collegato il terminale di trig. |
| Ultrasuoni(int, int) | Metodo costruttore | È il metodo costruttore della classe Ultrasuoni. È composto da due parametri: il primo esprime il pin di echo e il secondo il pin di trig. |
| distanza(): int | Metodo | Questo metodo ritorna la ditanza letta dall’ultrasuoni e quest’ultima viene convertita in centimetri. |

# Implementazione

## Design del circuito

Per iniziare abbiamo collegato il VCC del Digispark al terminale superiore del potenziometro, mentre il terminale inferiore è collegato ad un cavo che giunge alla massa. Il pin 2 del Digispark è collegato al terzo terminale che a sua volta è connesso al wiper (o tergicristallo) del potenziometro. Successivamente abbiamo congiunto il pin 0 del Digispark al anodo del led rosso, mentre il catodo giunge alla resistenza di 180Ω che a sua volta si reca alla massa.

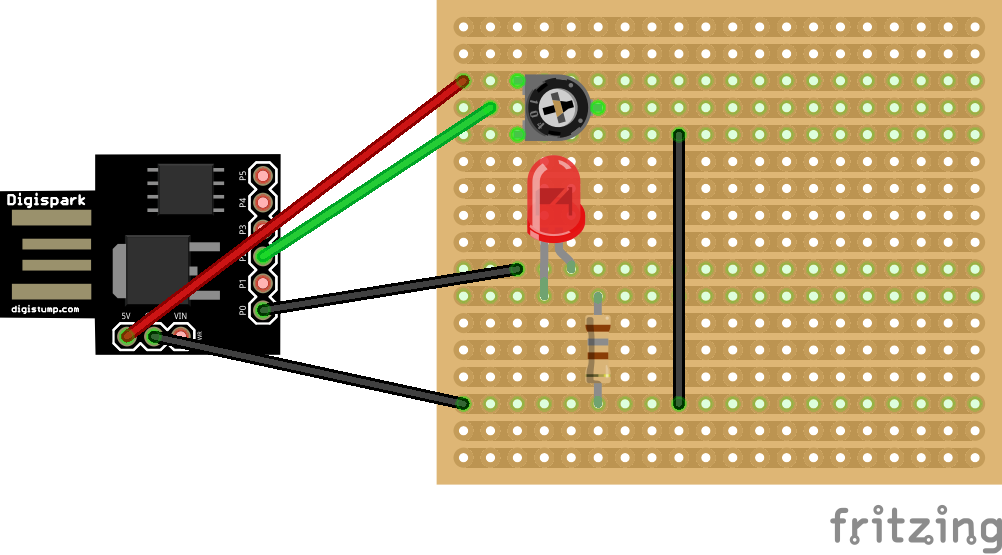


Figura 3, schema del circuito con potenziometro e led.

# 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

* <https://www.adrirobot.it/arduino/digispark/digispark.htm>, *Scheda Digispark*, 30-11-2018.
* <https://github.com/digistump/DigistumpArduino/releases/download/1.6.7/Digistump.Drivers.zip>, [*Digistump Arduino Release 1.6.7*](https://github.com/digistump/DigistumpArduino/releases/tag/1.6.7)*, 30-11-2018.*
* <https://www.instructables.com/id/Digispark-DIY-The-smallest-USB-Arduino/>, *Digispark DIY: the Smallest USB Arduino, 07-12-2018*
* <http://digistump.com/wiki/digispark>, *Getting Started with your Digispark or Digispark Pro, 12-12-2018.*
* <http://fritzing.org/download/>, *Fritzing, 12-12-2018.*

# 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
* …