Scanner Lidar

Documentazione

Sommario

[1 Introduzione 3](#_Toc71182398)

[1.1 Informazioni sul progetto 3](#_Toc71182399)

[1.2 Abstract 3](#_Toc71182400)

[1.3 Scopo 3](#_Toc71182401)

[2 Analisi 4](#_Toc71182402)

[2.1 Analisi del dominio 4](#_Toc71182403)

[2.2 Analisi e specifica dei requisiti 4](#_Toc71182404)

[2.3 Use case 6](#_Toc71182405)

[2.4 Pianificazione 7](#_Toc71182406)

[2.5 Analisi dei mezzi 8](#_Toc71182407)

[2.5.1 Software 8](#_Toc71182408)

[2.5.2 Hardware 9](#_Toc71182409)

[3 Progettazione 10](#_Toc71182410)

[3.1 Design dell’architettura del sistema 10](#_Toc71182411)

[3.2 Design dei dati 10](#_Toc71182412)

[3.3 Design delle interfacce 10](#_Toc71182413)

[3.4 Design procedurale 10](#_Toc71182414)

[4 Implementazione 10](#_Toc71182415)

[5 Test 11](#_Toc71182416)

[5.1 Protocollo di test 11](#_Toc71182417)

[5.2 Risultati test 11](#_Toc71182418)

[6 Consuntivo 12](#_Toc71182419)

[7 Conclusioni 12](#_Toc71182420)

[7.1 Sviluppi futuri 12](#_Toc71182421)

[7.2 Considerazioni personali 12](#_Toc71182422)

[8 Bibliografia 12](#_Toc71182423)

[8.1 Bibliografia 12](#_Toc71182424)

[8.2 Sitografia 12](#_Toc71182425)

[9 Allegati 12](#_Toc71182426)

# Introduzione

## Informazioni sul progetto

* Mandante: Geo Petrini.
* Partecipanti: Daniele Cereghetti, Isaac Gragasin, Veljko Markovic, Matteo Lupica.
* Tempo a disposizione: 14 gennaio 2021 - 20 maggio 2021
* Classe: I3AA-AC
* Progetto: Scanner Lidar

## Abstract

*Up until now, man strived to find various methods to acquire and analyse knowledge on many things, adapting and optimizing such procedures to be more efficient and effective as technological progress permitted so. This allowed the development of specialized scanning systems, fundamental for many professions, from modern doctors to simple shop clerks. Our project revolves around the use of a structure with a built-in LIDAR scanner and focuses on the extraction of data on surfaces in order to produce a graphic depiction of the current surroundings. A useful tool for jobs that are associated to architecture or that work heavily with planes, coordinates and metrics.*

## Scopo

Lo scopo del nostro progetto è quello di realizzare uno scanner LIDAR, utilizzando una combinazione tra una piattaforma Arduino e le capacità computazionali di un PC. Il quale dovrà visualizzare a schermo tutti i punti trovati dalla piattaforma.

# Analisi

## Analisi del dominio

È stato richiesto dal mandante di realizzare uno scanner LIDAR, utilizzando una combinazione tra una piattaforma Arduino e le capacità computazionali di un PC. Il prodotto sarà composto da due parti: una di Arduino, che si occupa di gestire l’aspetto meccanico e dell’inoltro dei dati al software; una, come accennato poco fa, di software, che si occupa invece di manipolare i dati ricevuti dalla parte fisica e di rappresentarli in uno spazio 3D. Il prodotto potrà girare solamente su sistemi operativi Windows 10 con architettura a 32 o 64 bit.

Potenzialmente, tutti potrebbero essere gli utenti di questo prodotto, ma è principalmente pensato per ambiti di architettura e misurazione. Le capacità minime richieste da parte degli utenti sono quelle di collegare l’Arduino ad un PC e di far partire il software.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-HW-1** | |
| **Nome** | Scansione a 360° sull’asse orizzontale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-HW-2** | |
| **Nome** | Scansione a 90° sull’asse verticale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-HW-3** | |
| **Nome** | Minor delay possibile tra scansione LIDAR e rotazione motore |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-HW-4** | |
| **Nome** | Invio dati al PC |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Inviare l’angolo della rotazione orizzontale, verticale e la distanza rilevata |

|  |  |
| --- | --- |
| **ID: REQ-SW-1** | |
| **Nome** | Acquisire i dati da Arduino tramite USB |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-SW-2** | |
| **Nome** | Interpretare i dati in ambiente 3D |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Trasformare i valori ottenuti dalla scansione in vettori |

|  |  |
| --- | --- |
| **ID: REQ-SW-3** | |
| **Nome** | Creazione dell’applicativo per un solo sistema operativo |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Sistema operativo scelto: Windows |

|  |  |
| --- | --- |
| **ID: REQ-SW-4** | |
| **Nome** | L’applicativo deve essere un file eseguibile |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |

|  |  |
| --- | --- |
| **ID: REQ-SW-5** | |
| **Nome** | Linguaggio di programmazione da utilizzare |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Linguaggio scelto: C# |

## Use case

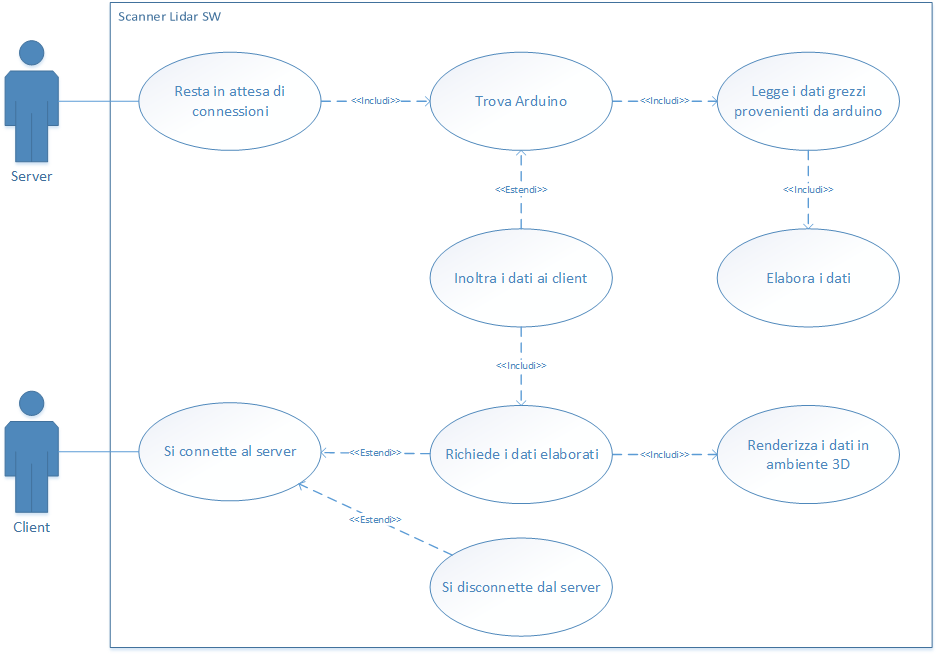


Figura - Use Case

## Pianificazione

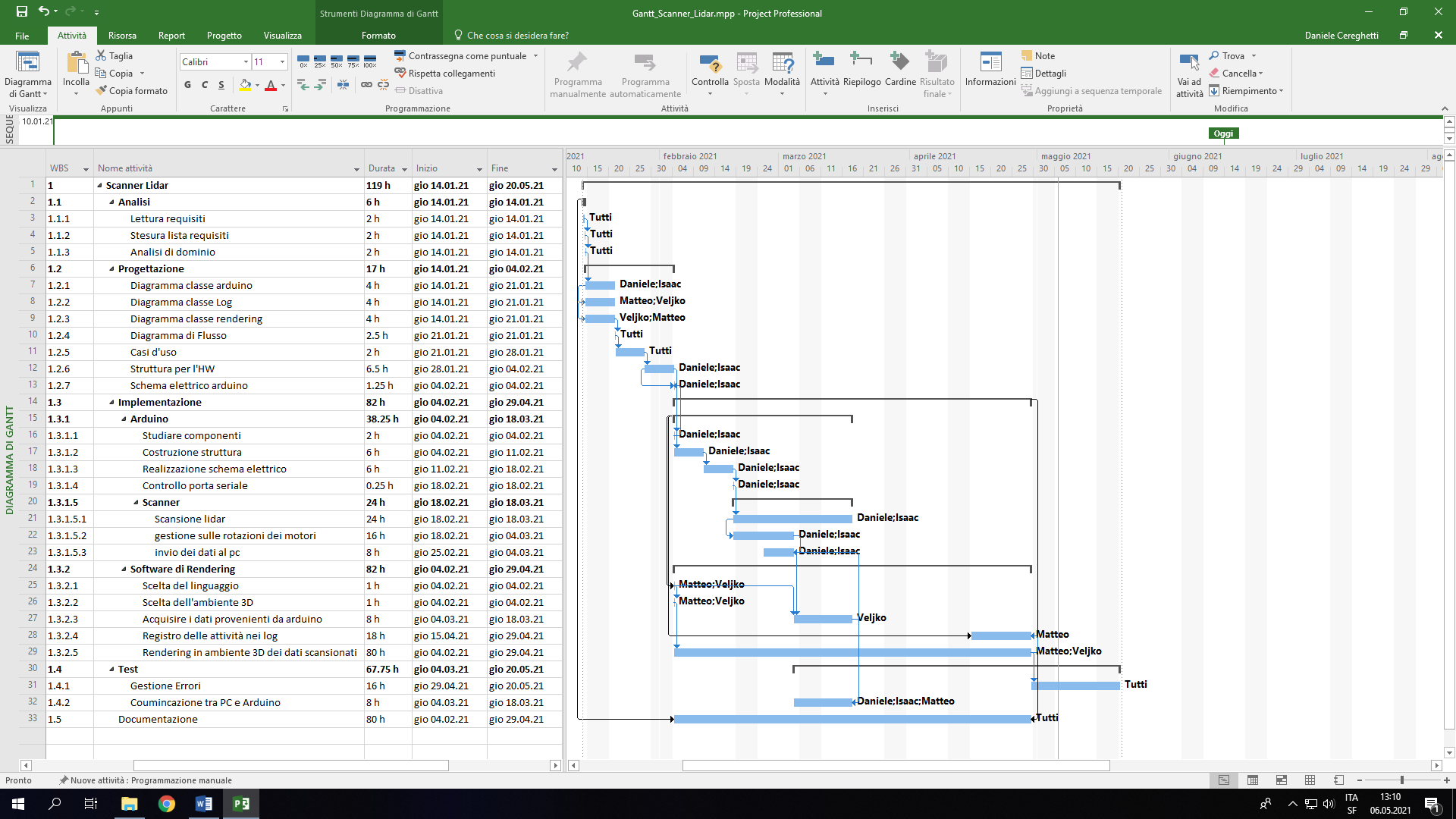


Figura - Gantt preventivo

## Analisi dei mezzi

### Software

### Tools

Per la realizzazione del server è stato utilizzato:

* Visual Studio 2019 (16.7.2)

Per la realizzazione del client sono stati utilizzati:

* Visual Studio 2019 (16.8.6)
* Unity (2020.2.3.f1 Personal <DX11>)

Per la realizzazione del codice di Arduino è stato utilizzato:

* Arduino IDE (1.8.13)

Per la realizzazione della struttura dello scanner Lidar è stato utilizzato:

* LeoCAD (21.01)
* Easel (WebApp)

Per la realizzazione dei diagrammi e use case sono stati utilizzati:

* Draw IO (WebApp)
* Microsoft Visio 2016

Per la realizzazione del Gantt è stato utilizzato:

* Microsoft Project 2016

Per la realizzazione della documentazione è stato utilizzato:

* Microsoft Word 2016

Per la creazione e l’utilizzo della macchina virtuale è stato utilizzato:

* Oracle VM VirtualBox (6.1.16 r140961 (Qt5.6.2))

### Librerie

### Arduino

Le librerie elencate di seguito sono state utilizzate per la realizzazione della parte di Arduino e sono disponibili sul nostro repository di GitHub:

* Arduino JSON (6.17.3)
* TFmini Arduino (0.1.1)

### Server

Le librerie utilizzate per la realizzazione del Server sono:

* Microsoft.EntityFrameworkCore (3.1.12)
* Microsoft.EntityFrameworkCore.Sqlite (3.1.12)
* Microsoft.EntityFrameworkCore.Tools (3.1.12)
* Microsoft.Extensions.Hosting (3.1.12)
* NLog (4.7.7)
* NLog.Config (4.7.7)
* System.Configuration.ConfigurationManager (5.0.0)
* System.IO.Ports (5.0.0)
* System.Management (5.0.0)

### Hardware

Il prodotto dovrà essere eseguito su una macchina Windows 10.

### Arduino e le sue componenti

Per la realizzazione della struttura fisica dello scanner Lidar, sono stati utilizzati:

* Arduino Mega 2560
* TFmini-S LiDAR Module
* Due STEP MOTOR 28BYJ-48 5VDC
* Driver ULN2003

### Macchina Virtuale

Per problemi di proxy, non abbiamo potuto installare Unity sui PC fisici di scuola. Pertanto, abbiamo optato di installare e utilizzare Unity su una macchina virtuale con le seguenti specifiche di sistema:

* Sistema operativo: Windows 10 Home (1909) con architettura a 64-bit
* Processore: 2 Core
* RAM: 8 GB
* Memoria: 50 GB

Per quanto concerne le impostazioni della macchina virtuale, sono le seguenti:

* Scheda di rete: NAT
* Audio:
  + Uscita: Sì
  + Entrata: No
* USB: Abilitato

### Macchina fisica scolastica

I PC che abbiamo utilizzato e che avevamo a disposizione a scuola hanno le seguenti specifiche:

* Sistema operativo: Windows 10 Education (1803) con architettura a 64-bit
* Processore: Intel® Core™ i7-7700 CPU @ 3.60 GHz
* RAM: 16 GB
* Memoria: 460 GB

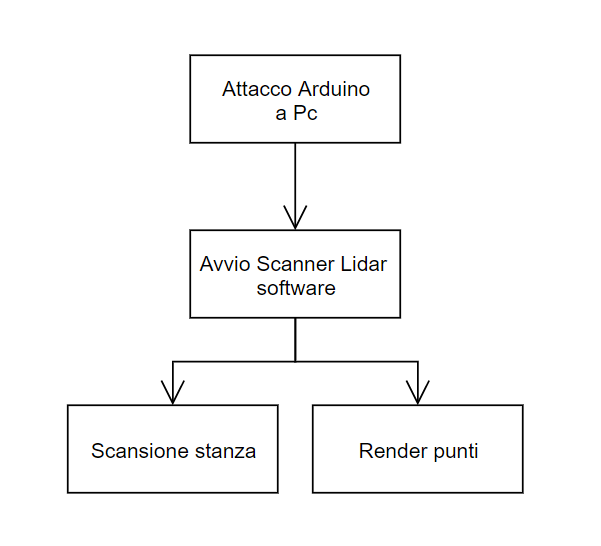
### Portatile

Per la realizzazione della parte di Arduino, abbiamo utilizzato un portatile MacBook Pro (Retina, 15-inch, Mid 2015) con le seguenti specifiche:

* Sistema operativo: macOS Big Sur (11.0.1)
* Processore: Intel® Core™ i7 CPU @ 2.50 GHz Quad-Core
* RAM: 16 GB
* Memoria: 250 GB

# Progettazione

## Design dell’architettura del sistema



<Spiegare cose>

## Design dei dati

## Design delle interfacce

## Design procedurale

# Implementazione

## Server

Il ruolo del server all’interno del progetto e quello di elaborare i dati provenienti da Arduino, gestire il log, gestire i client e distribuirgli i dati necessari.

Per poter leggere i dati provenienti sulla seriale abbiamo bisogno di sapere su quale porta comunica Arduino. Dopo che sappiamo su che porta comunica, creiamo un buffer grande quanto i dati disponibili sulla seriale e poi convertiamo e leggiamo il suo valore.

byte[] buffer = new byte[arduino.BytesToRead];

string reader = "";

for (int i = 0; i < buffer.Length; i++){

reader += (char)arduino.ReadByte();

}

## Client

## Arduino

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | 1  1 | **Descrizione:** | Software che fa il cambio di ruolo con il server ed il client |
| **Prerequisiti:** | Nessuno | | |
| **Procedura:** | 1. Aprire il terminale, andare sotto la cartella nella quale si trova il programma ed eseguire il programma (java LanSpeedTester) 2. Inserire il numero che corrisponderà al ruolo del server (1) oppure il client (2) | | |
| **Risultati attesi:** | Caso 1 (server):   * Vengono richiesti i parametri per il server   Caso 2 (client):   * Vengono richiesti i parametri per il client | | |

## Risultati test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | 1  1 | **Descrizione:** | Software che fa il cambio di ruolo con il server ed il client |
| **Risultati:** | Caso 1 (server):   * Vengono richiesti i parametri per il server   Caso 2 (client):   * Vengono richiesti i parametri per il client | | |
| **Risultato finale:** | Positivo | | |

# Consuntivo

# Conclusioni

## Sviluppi futuri

## Considerazioni personali

# Bibliografia

## Bibliografia

## Sitografia

# Allegati