**1 INTRODUZIONE**

**1.1 Obiettivo**

La stesura di tale documento ha come obiettivo quello di illustrare i requisiti richiesti e l’architettura hardware e software di un dispositivo di riconoscimento di colori. Tale dispositivo è realizzato con l’idea di facilitare in qualche modo la percezione dei colori degli oggetti che ci circondano di coloro che abbiano difficoltà visive, in particolare non vedenti e/ daltonici.

**1.2 Scopo**

Il dispositivo **ReColor** deve essere in grado di identificare il colore dell’oggetto in prossimità del quale viene posto, riuscendo inoltre ad associare al colore identificato una segnalazione vocale che metta a conoscenza all’utente del colore dell’oggetto in questione.

**1.3 Struttura generale**

Nel paragrafo 2 viene effettuata una descrizione delle funzionalità del sistema, elencando anche quelli che sono i requisiti funzionali e non funzionali del sitema. Il paragrafo 3 invece descrive le funzionalità del dispositivo sotto forma di casi d’uso. Inoltre viene riportata una descrizione dell’architettura sia hardware che software del sistema. Infine, il paragrafo 4 riporta le tecniche di testing utilizzate.

**2 Requisiti del sistema**

**2.1 Interfaccia utente**

Il dispositivo è dotato di un pulsante che permette una volta avvicinato il sensore di riconoscimento di colori all’oggetto in questione di rilevare il colore del suddetto oggetto, quindi inviare il segnale al sistema che sarà in grado di trasmettere il file audio corrispondente. Un led resta acceso in condizioni di standby del sistema mentre si spegne durante la segnalazione acustica.

**2.2 Caratteristiche utente**

Gli utenti destinatari del prodotto **ReColor** sono coloro i quali presentino difficoltà visive e non riescano a distinguere i colori. Potrebbero inoltre farne uso tutti colori i quali volessero riconoscere il colore dell’oggetto in riferimento.

**2.3 Vincoli**

* L’utente può mettere in funzione il dispositivo semplicemente pigiando il pulsante di cui è fornito.
* Ambienti di sviluppo: IDE di Arduino.
* Il dispositivo è provvisto di una microSD per la memorizzazione dei file audio di una capienza pari a 1GB.
* Il modulo che permette l’utilizzo della microSD non supporta memorie superiori ai 2 GB.
* La distanza tra l’oggetto ed il dispositivo non deve essere superiore ai 10 mm.

**2.4 Requisiti funzionali**

Sono stati fissati i seguenti requisiti funzionali:

* Il dispositivo dispone di un pulsante che permette l’esecuzione del sistema;
* Il dispositivo permette di rilevare tramite modulo sensore di riconoscimento colori i valori di intensità luminosa di rosso, verde e blu;
* Il dispositivo dispone di un modulo di memoria dei file audio necessario ad associare al colore rilevato la segnalazione acustica corrispondente;
* Il dispositivo dispone di un led che permette di segnalare se il sistemaa è o meno in funzionamento;

**2.5 Requisiti non funzionali**

Sono stati fissati i seguenti requisiti non funzionali:

1. **Documentazione:**

* Piano di Software Configuration Management;
* Analisi dettagliata dei requisiti;
* Source Code;

La suddetta documentazione è stata gestita utilizzando come sistema di versioning GitHub ed il repository è consultabile al link: https://github.com/NdiVico/Recognizer\_colors.

1. **Ambiente fisico di operatività:**

Il sensore può operare in qualsiasi ambiente fisico, però la corretta identificazione dei colori è garantita solo per oggetti con determinati vincoli di riflettività. ??????

**3 Modellazione del sistema**

**3.1 Specifiche dei Casi d’uso**

I casi d’uso sono impiegati per descrivere il comportamento “esterno” del sistema da sviluppare, senza dover specificare come tale comportamento viene realizzato. Forniscono una descrizione dei modi in cui il sistema potrà essere utilizzato. Essi sono basati sulla metafora del dialogo tra utente (attore) e sistema. L’attore rappresenta qualsiasi elemento esterno al sistema (persona o altro sistema) che usa il sistema per fare qualcosa. Nel nostro caso l’attore è l’utente e il sistema è Arduino.

Un caso d’uso è una collezione di scenari, di successo o di fallimento del conseguimento dell’obiettivo, correlati tra loro che descrivono come un attore usa un sistema per raggiungere un obiettivo.

Di seguito è riportato il caso d’uso, con i relativi scenari, riferito al sistema realizzato:

**UC1**

* **Caso d’uso:** Controllo tramite button del sistema
* **Attore:** Utente.
* **Input:** Pressione del botton posto sul dispositivo.
* **Precondizioni:** Il dispositivo è dotato di un led che permette di segnalare se il sistema è o meno in funzionamento. Inoltre esso deve essere disposto in modo che il modulo di riconoscimento dei colori sia sufficientemente vicino all’oggetto del quale si vuole riconoscere il colore.
* **Output:** 
  1. (scenario 1) Il dispositivo riconosce il colore, (1) l’altoparlante enuncia il colore rilevato.
  2. (scenario 2) Il dispositivo non riconosce il colore, (2) “colore non rilevato”.
* **Post Condizione:** Il button è rilasciato ed il led è acceso.
* **Descrizione scenario:**
  + - * 1. **Scenario 1**

**1.a** L’utente avvicina il sensore di riconoscimento di colori all’oggetto.

**1.b** L’utente preme il bottone presente sul dispositivo.

**1.c** Il dispositivo acquisisce e riconosce il colore dell’oggetto.

**1.d** L’altoparlante emette il file audio corrispondente al colore rilevato (1) ed il led si spegne.

* + - * 1. **Scenario 2**

**1.a** L’utente avvicina il sensore di riconoscimento di colori all’oggetto.

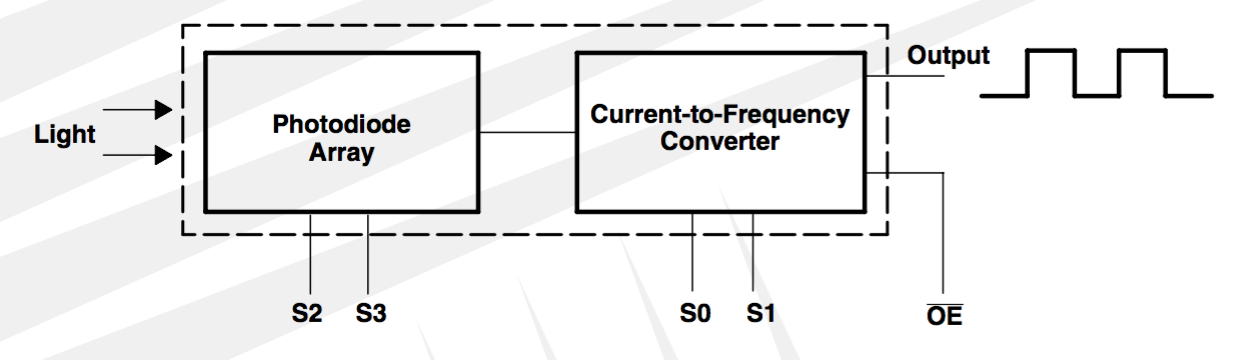
**1.b** L’utente preme il bottone presente sul dispositivo.

**1.c** Il dispositivo acquisisce e non riconosce il colore dell’oggetto.

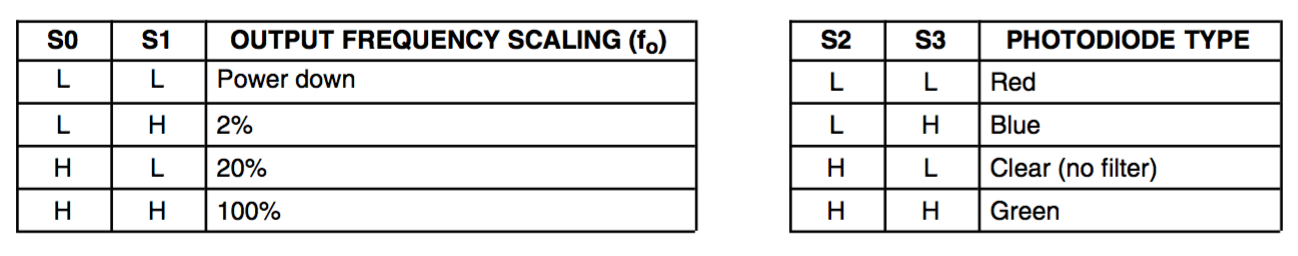
**1.d** L’altoparlante emette il messaggio (2).

**3.2 Descrizione Hardware del sistema**

**3.2.1 Modulo TCS230**

****

Il modulo TCS230 è un convertitore colore – luce – frequenza. Quest’ultimo è composto da 16 fotodiodi con filtri rossi, 16 fotodiodi con filtri blu, 16 fotodiodi con filtri verdi e 16 non filtrati. Tutti i fotodiodi dello stesso colore sono posti in parallelo. Quando un oggetto è posto frontalmente al modulo, ad una distanza non superiore ai 10 mm, questo viene illuminato dai 4 led bianchi e la luce riflessa andrà a colpire i 64 fotodiodi, ottenendo in uscita un’onda quadra (Duty cycle 50%) con una frequenza direttamente proporzionale all’intensità della luce riflessa. La frequenza a fondoscala in uscita può essere regolata su uno dei tre valori preimpostati disponibili tramite 2 piedini dell’ingresso di controllo (S0 e S1) come riportato in Tab.1. Gli ingressi digitali e l’uscita digitale permettono di interfacciarlo direttamente ad arduino o ad altri circuiti logici. I piedini S2 e S3 sono usati per selezionare quale gruppo di fotodiodi (rosso, verde, blu, chiaro) sono attivi.



**3.2.2 Modulo WTV020SD-16p**

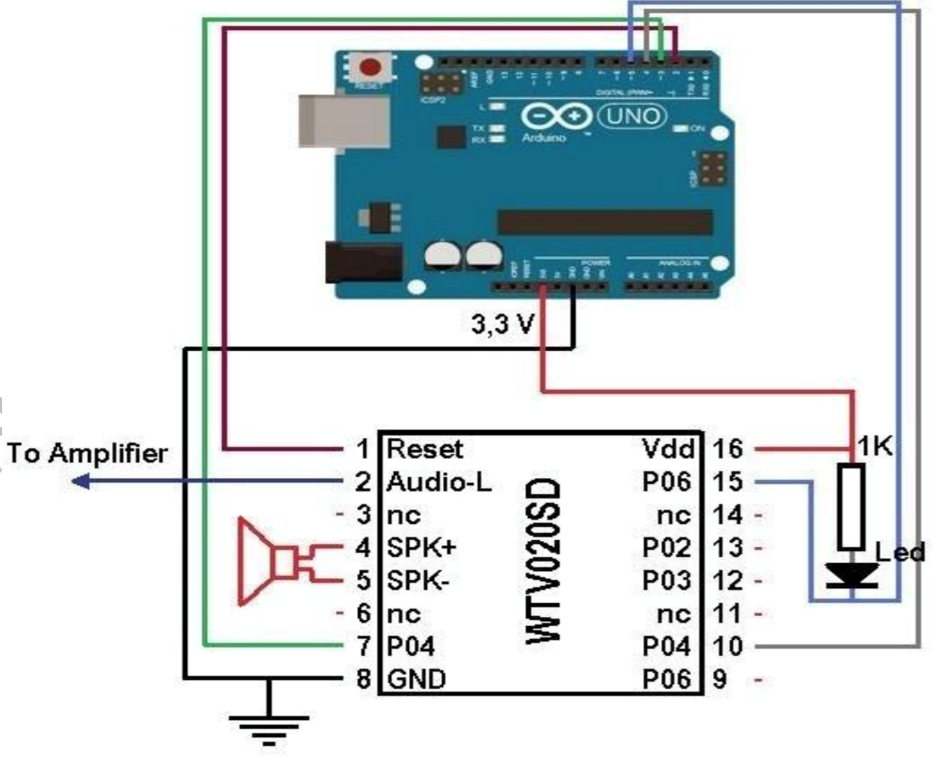
Il modulo WTV020SD-16p ci permette di riprodurre dei brani salvati su una micro SD, e grazie all’interfaccia con Arduino è possibile scegliere quale brano eseguire ed in quale momento. Con questo modulo possiamo lavorare in due modi:

* Autonomamente (con l’ausilio di un piccolo circuito).
* Arduino (mediante l’utilizzo di un’opportuna libreria).

Può essere collegato direttamente ad un piccolo altoparlante o ad un amplificatore.

Tale dispositivo non legge direttamente file .mp3 ma file .ad4, quindi prima di caricarli sulla micro SD è necessario convertirli con un tool opportuno.

La scelta della scheda micro SD è molto critica, infatti accetta schede con capacità massima di 2 Gb e non tutte funzionano.



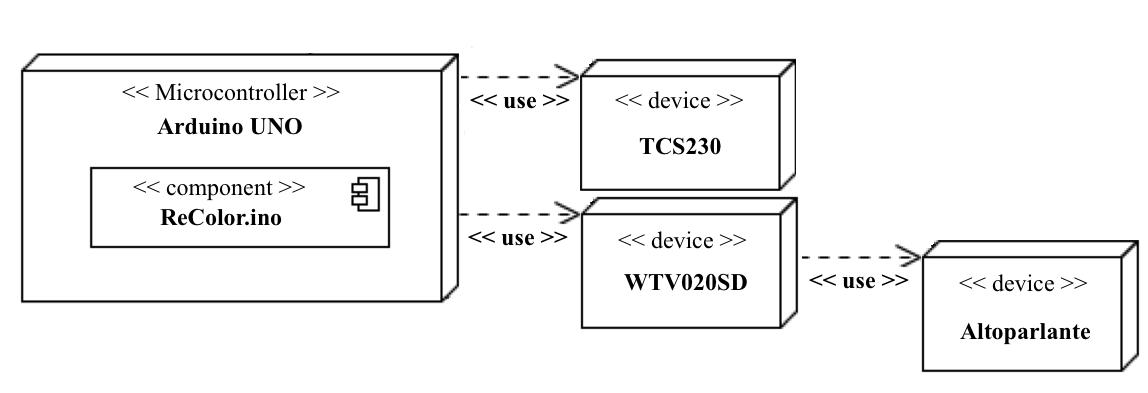
**3.3 Flow Chart del codice Arduino**

**3.4 Diagrammi UML**

**3.5 Deployment diagram**

Un deployment diagram viene sfruttato per modellare l’hardware utilizzato per l’implementazione del sistema e i collegamenti tra i diversi pezzi hardware. L’UML fornisce dei simboli che hanno il compito di favorire la creazione di un quadro chiaro di come il settaggio e la composizione finale dell’hardware dovrà essere.

Un cubo rappresenta un nodo che corrisponde ad un elemento fisico del sistema. Un nodo ha un suo nome ed è possibile anche usare uno stereotipo per indicare il tipo di risorsa che esso rappresenta. Se un nodo fa parte di un package, allora il nome del package deve precedere il nome del nodo. Una linea che unisce due cubi rappresenta una connessione tra i due nodi. E’ possibile usare uno stereotipo anche per fornire informazioni sulla connessione. Essi possono anche comprendere ulteriori componenti del sistema al proprio interno, quali ad esempio le componenti software utilizzate. In Fig.1 viene riportato il deployment diagram relativo all’architettura hardware del progetto:

****

Per lavorare correttamente Arduino, stereotipato come “microcontrollore”, ha bisogno di usufruire di diversi device esterni ovvero il trasduttore TCS230 ed il modulo ausiliario WTV020SD. Quest’ultimo è interconnesso all’altoparlante.

**4 Testing**