

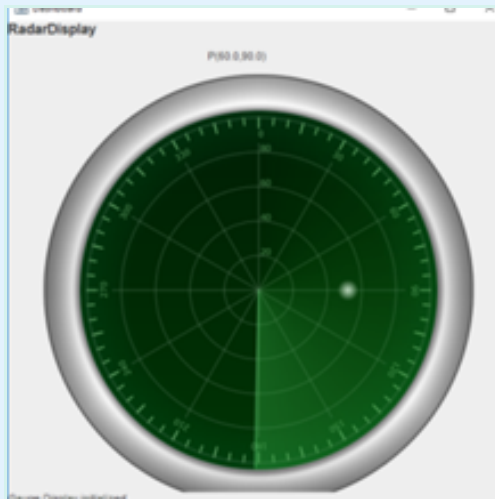
LABORATORIO DI INGEGNERIA DEI SISTEMI SOFTWARE

Introduction

Requirements

Si desidera costruire un'applicazione software capace di:

- (requisito radarGui) mostrare le distanze rilevate da un sensore Sonar "HC-SR04" connesso a un RaspberryPi su un display (RadarDisplay) a forma di radar connesso a un PC.



- (requisito ledAlarm) accendere un Led se la distanza rilevata dal Sonar è inferiore a un valore limite prefissato denominato DLIMIT.

Requirement analysis

Problem analysis

Aspetti Rilevanti

1. Si tratta di realizzare un sistema distribuito formato da quattro componenti:
 - Il Sonar HC-SR04, di cui il committente fornisce il seguente software [SonarAlone.c](#)
 - Il RadarDisplay, di cui è disponibile il POJO realizzato da radarPojo.jar
 - Il Led, di cui il committente fornisce codice di basso livello come [Led25GpioTurnOn.sh](#) e [Led25GpioTurnOff.sh](#)
 - Il Controller, che ha il compito di ricevere in ingresso i dati del sensore HC-SR04, elaborarli e

inviare comandi al Led e dati al RadarDisplay

2. L'interazione sonor-radar puo' avvenire in due modi:

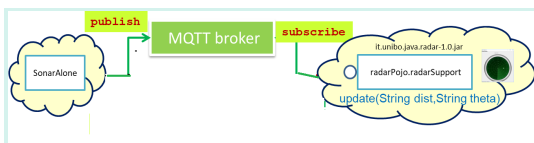
- **Comunicazione diretta**

La figura rappresenta gli strati di software che permettono ai dati generati dal Sonar di essere ricevuti dal RadarDisplay



- **Comunicazione mediata**

Richiede la presenza di un componente mediatore, realizzato da terze parti come servizio disponibile

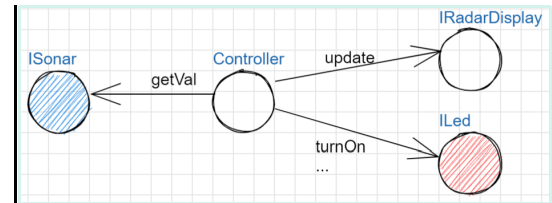


3. Per rendere comprensibile l'architettura alla 'macchina', introduciamo opportuni modelli dei componenti utilizzando un linguaggio di programmazione. In particolare, nel caso di Java, è possibile utilizzare il costrutto interface per definire un componente catturando le funzionalità che esso deve offrire. A consunto di ciò possiamo implementare le seguenti interfacce:

Led	RadarDisplay
<pre> public interface ILed { public void turnOn(); public void turnOff(); public boolean getState(); } </pre>	<pre> public interface IRadarDisplay{ public void update(String d, String a); } </pre>
Distance	Sonar
<pre> public interface IDistance { public int getVal(); public String toString(); } </pre>	<pre> public interface ISonar { public void activate(); public void deactivate(); public IDistance getDistance(); public boolean isActive(); } </pre>

Architettura Logica

Un primo schema dell'architettura logica può essere definito come mostrato in figura



Test plans

Project

Testing

Deployment

Maintenance

By Alessio Reitano email: alessio.reitano@studio.unibo.it

