

Smart radar

Alberto Marfoggia, Elia Marcantognini, Tommaso Mandoloni

21 novembre 2019

Indice

1	Hardware	3
1.1	Schema del circuito	3
1.2	Lista dei sensori e attuatori utilizzati	3
2	Software	4
2.1	Scelte implementative	4
2.2	Tasks	4
2.2.1	ManualModeTask	4
2.2.2	AutoModeTask	5
2.2.3	SingleModeTask	6
2.2.4	BlinkingTask	7
2.2.5	ChangingModeTask	8
2.3	Smart Radar Manager - GUI	8

1 Hardware

1.1 Schema del circuito

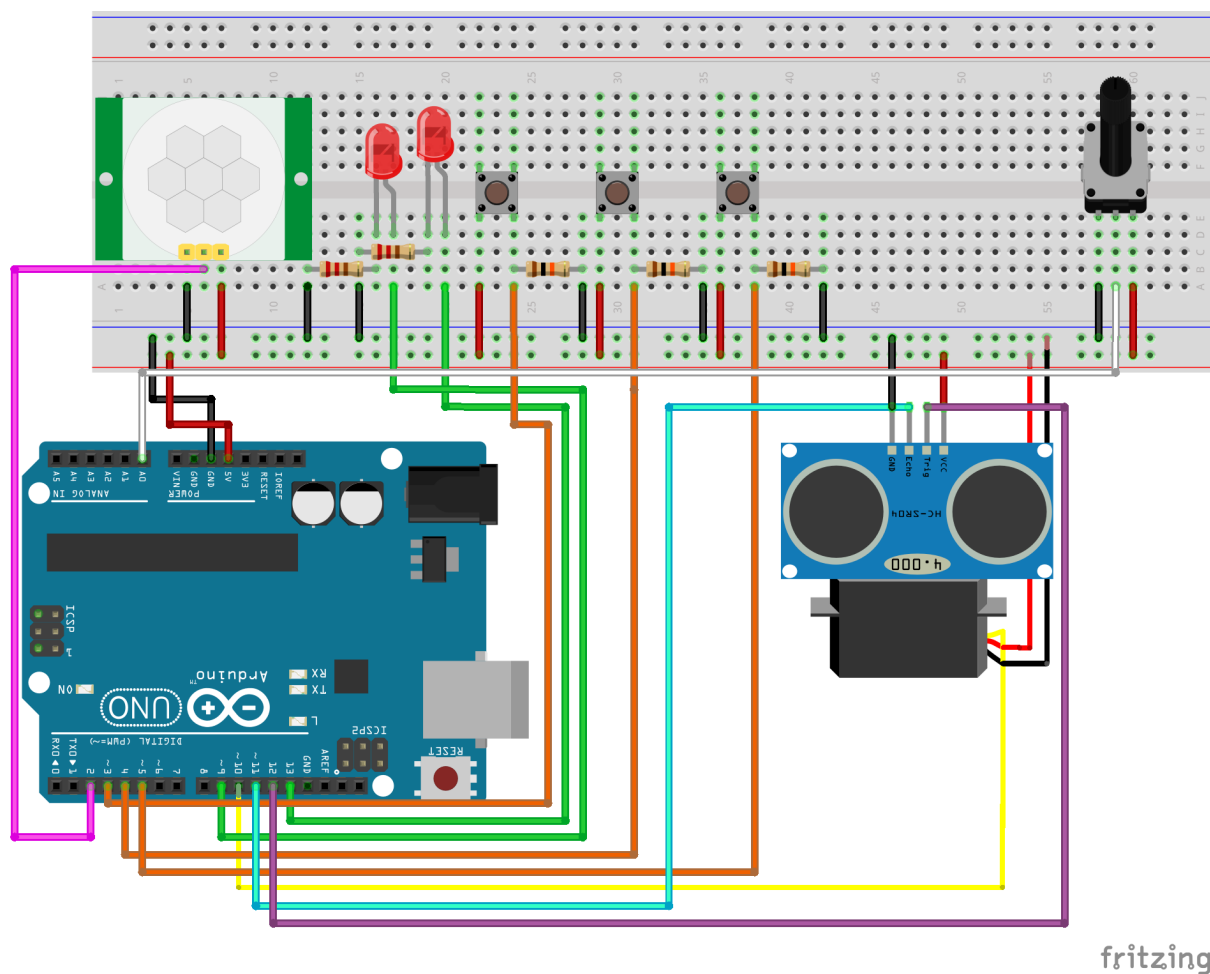


Figura 1: Schema del circuito realizzato mediante Fritzing.

1.2 Lista dei sensori e attuatori utilizzati

- PIR: utilizzato per il rilevamento di movimenti nell'ambiente esterno. Ha un ruolo fondamentale nella gestione del risparmio energetico del sistema.
- Sonar: utilizzato per effettuare le scansioni e calcolare la distanza che c'è tra il dispositivo e l'oggetto rilevato.
- Servo: funzione esclusivamente meccanica, permette al sonar di compiere rotazioni di 180 gradi.
- Potenziometro: utilizzato per ridurre o aumentare la frequenza di scansionamento.
- Led: utilizzati per notificare il rilevamento di un oggetto.
- Pulsanti: utilizzati per il cambio di modalità.

2 Software

2.1 Scelte implementative

In fase di progettazione, si è scelto di scomporre il dominio applicativo in sottoparti per rendere più agevole lo sviluppo di quest'ultime. Seguendo questa soluzione si è realizzato un task per il sistema di allarme, un task per ogni modalità (auto, single, manual) ed un ulteriore per la gestione del passaggio da una modalità all'altra. Fondamentale è stata l'implementazione di uno scheduler dinamico tramite il quale è possibile abilitare o disabilitare in qualsiasi momento un task. Abbiamo deciso di utilizzare questa tipologia di scheduler poiché permette di ottimizzare i tempi di calcolo, visto che si perdevano diversi cicli di clock per interrogare le tre modalità ed identificare quella attiva. Alcuni task dovevano cambiare stato in base ad azioni invocate da altri, si è deciso di implementare un oggetto condiviso chiamato StateManager preferendolo ad una classe statica globale (Singleton) poiché non strettamente necessaria.

2.2 Tasks

2.2.1 ManualModeTask

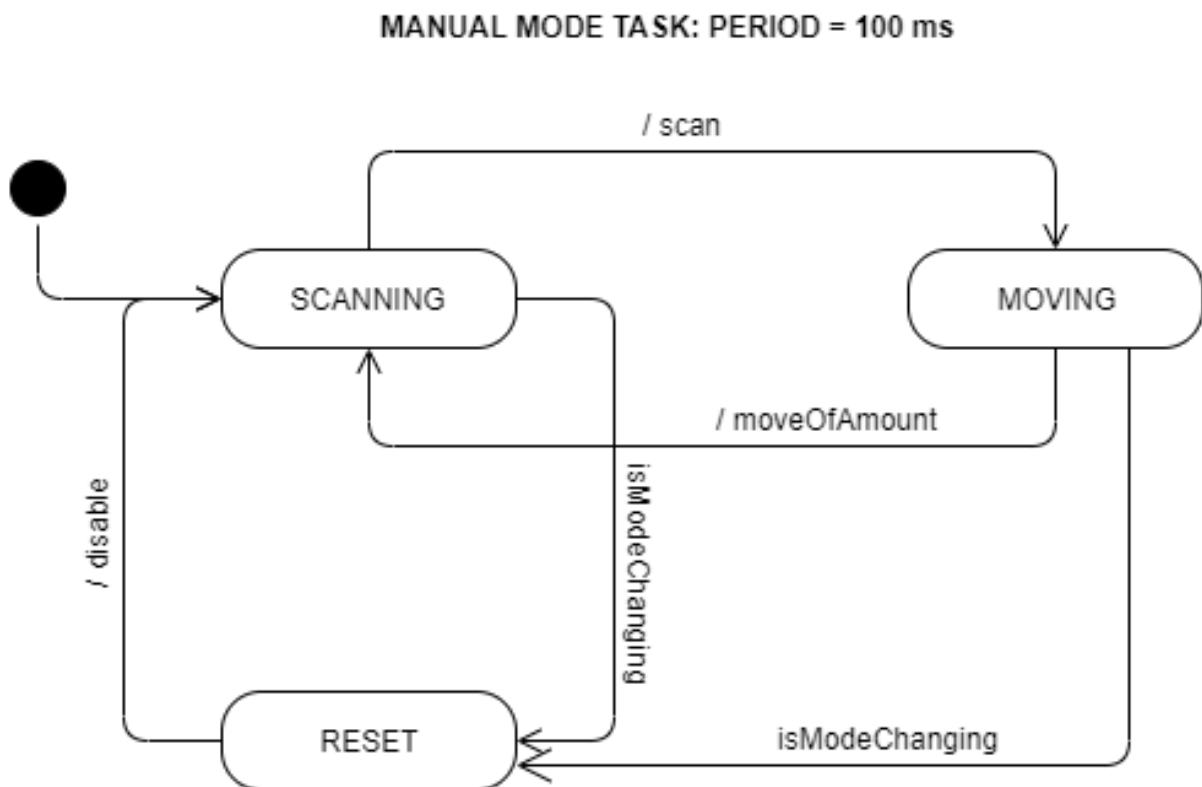


Figura 2: Diagramma a stati finiti del ManualModeTask

La Manual Mode è la modalità con cui viene avviato il sistema dello Smart Radar. È completamente controllata da PC mediante l'invio di comandi su seriale da un programma Java ed utilizza solamente il Radar, che si compone di servo e sonar. La sua funzione principale è quella di effettuare scansioni che consistono nel notificare direzione e distanza di un oggetto, se rilevato, in base agli spostamenti richiesti dall'utente tramite console.

2.2.2 AutoModeTask

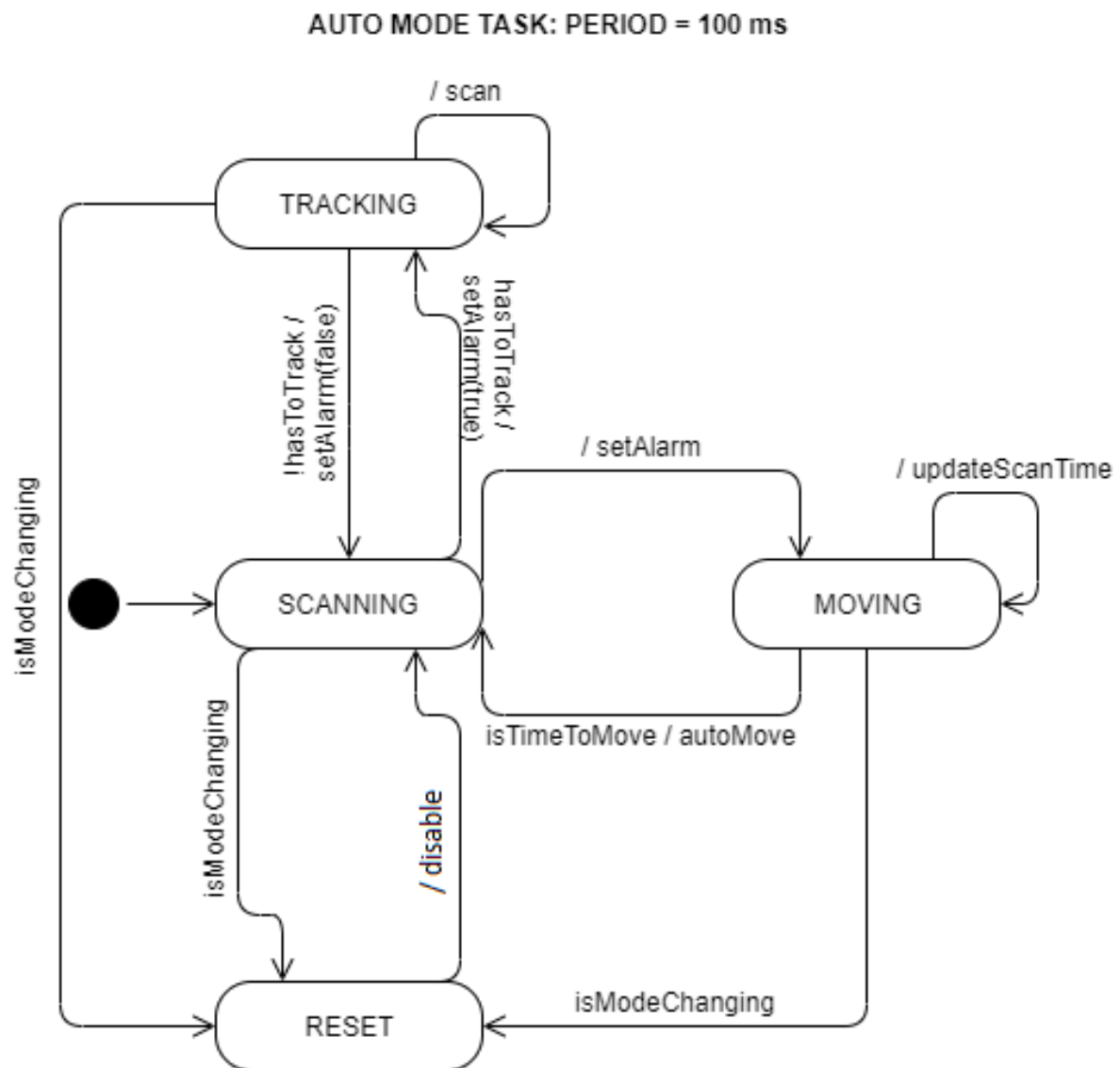


Figura 3: Diagramma a stati finiti dell'AutoModeTask

L'Auto Mode è la modalità che non richiede interazione con l'utente. A differenza della Manual quando il radar rileva un oggetto entro un certo intervallo deve entrare in "stato di allarme" sfruttando i metodi forniti dallo StateManager. Inoltre in questa modalità è possibile modificare la frequenza di scansionamento tramite potenziometro o console.

2.2.3 SingleModeTask

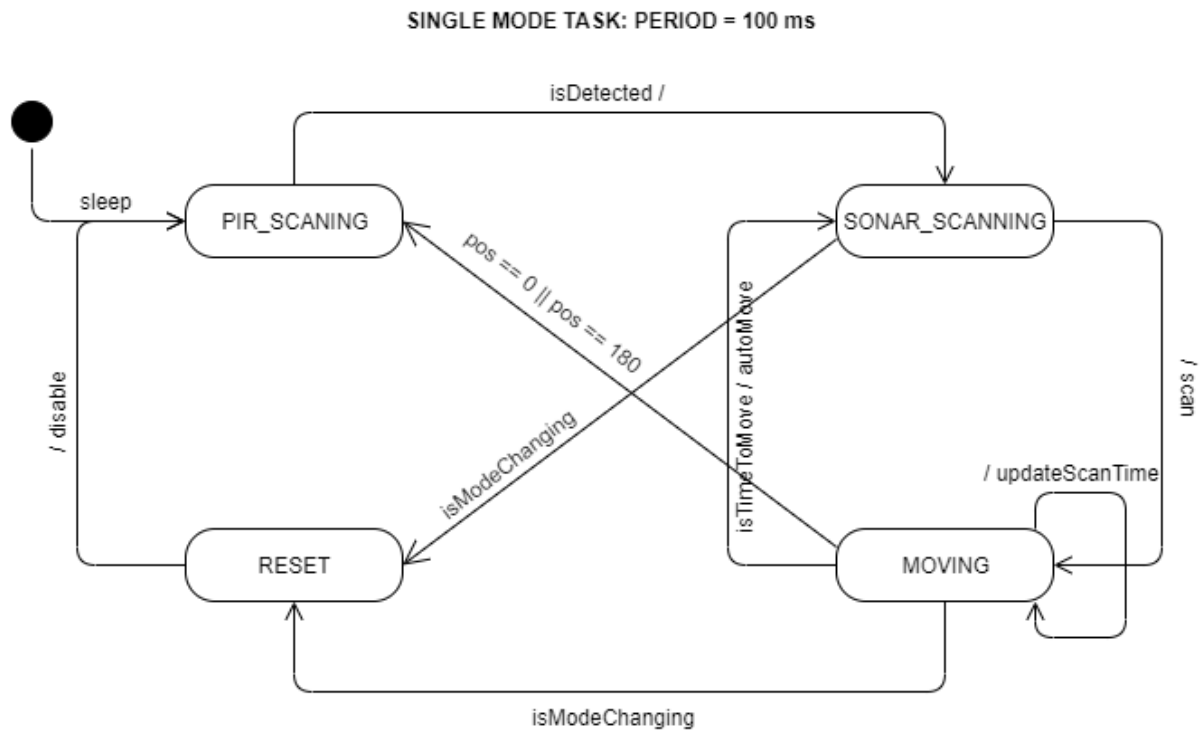


Figura 4: Diagramma a stati finiti del SingleModeTask

Nella Single Mode Task, rispetto alla modalità automatica, viene utilizzato anche il PIR. Inizialmente il sistema entra in standby, successivamente, se il PIR rileva un movimento, il sistema viene risvegliato tramite interrupt e inizia ad eseguire una scansione. In questo modalità quando il radar rileva un oggetto a una certa distanza, il task utilizza lo StateManager per ordinare al BlinkingTask di effettuare un semplice blink.

2.2.4 BlinkingTask

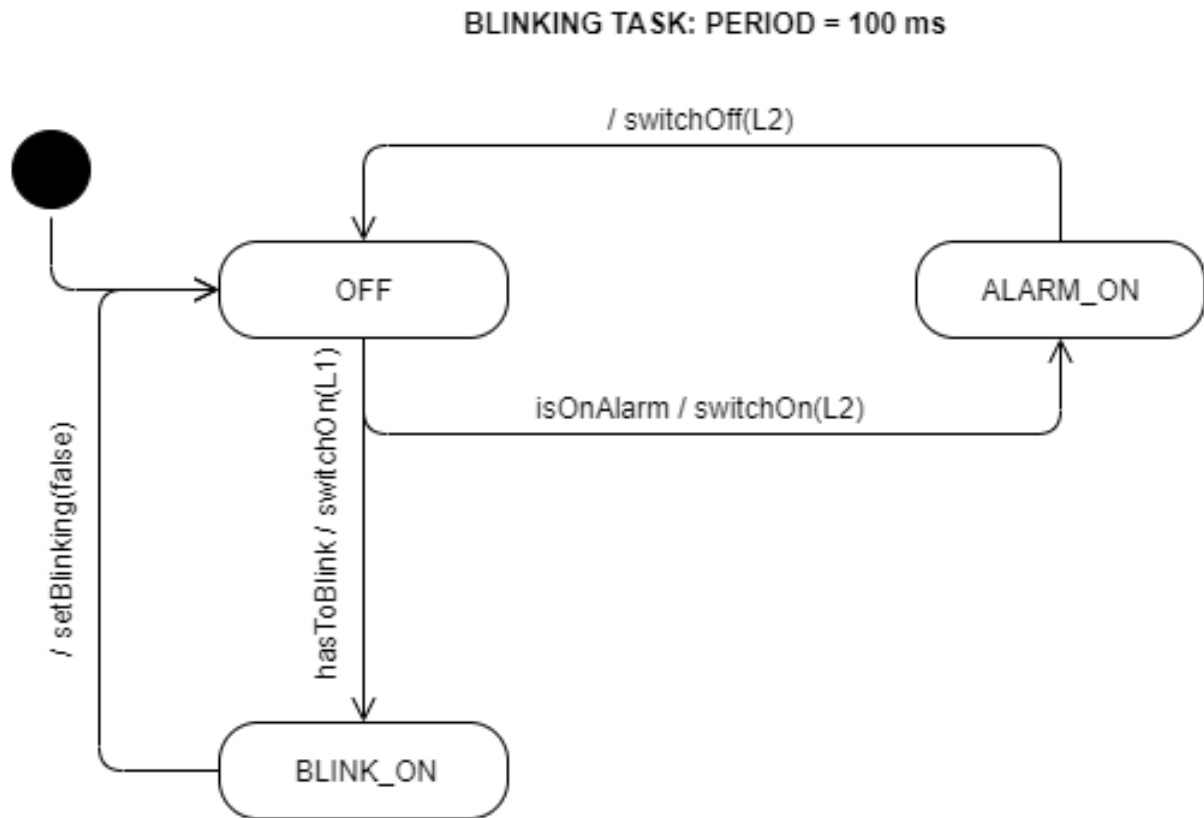


Figura 5: Diagramma a stati finiti del BlinkingTask

Il BlinkingTask è stato implementato con lo scopo di gestire sia il sistema di allarme, rappresentato da un blinking continuo, sia un semplice blink. Il primo viene utilizzato dall'Auto task mentre il secondo dal Single per notificare il rilevamento di un oggetto. In entrambi i casi le guardie sono definite da determinati metodi dello StateManager. Dato che i due led sono legati a due differenti modalità le quali non possono sovrapporsi, è stato sufficiente gestire l'allarme e il blink mediante due stati distinti.

2.2.5 ChangingModeTask

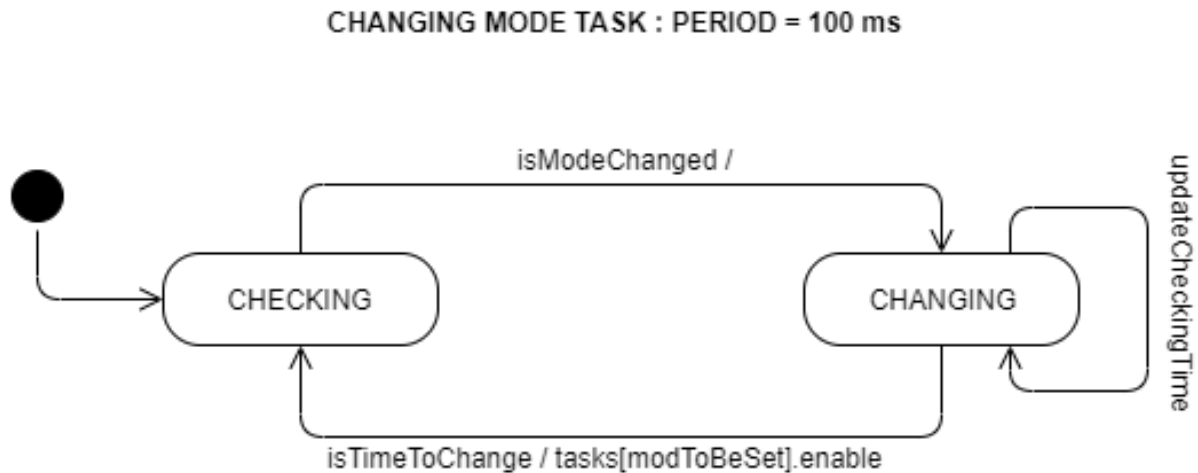


Figura 6: Diagramma a stati finiti del ChangingModeTask

Il ChangingModeTask realizza un sistema polling impiegato nel rilevamento di richieste di cambio modalità, le quali possono essere ricevute mediante console o pressione di uno dei tre pulsanti installati nel circuito. Quando arriva una richiesta il task non cambia subito modalità ma entra in uno stato transitorio chiamato CHANGING dal quale esce solamente dopo un certo periodo necessario per disabilitare il task relativo alla modalità corrente e per riposizionare meccanicamente il radar. Si è scelto un approccio polling perchè il numero di pin interrupt della scheda erano insufficienti e perchè la problematica del bouncing è stata risolta semplicemente scegliendo un opportuno periodo del task.

2.3 Smart Radar Manager - GUI

Lo Smart Radar Manager è il sistema software scritto in Java ed eseguibile su PC utilizzabile per gestire il funzionamento dello Smart Radar. Per garantire il funzionamento di questo sistema ci siamo avvalsi della libreria JSSC grazie alla quale siamo riusciti ad instaurare una comunicazione seriale con il microcontrollore. A questo proposito, abbiamo deciso di creare un "protocollo" con messaggi predefiniti per garantire il corretto funzionamento dell'integrazione tra i due sistemi. È composto dai seguenti componenti:

- **MonitoringAgent:** è il componente che ti occupa della ricezione di messaggi inviati dallo SmartRadar attraverso la Seriale (gestita in SerialCommChannel).
- **SmartRadarView e LoggerView:** componenti che rappresentano la GUI principale. In particolare:
 - **SmartRadarView:** rappresenta il componente con pulsanti per gestire il sistema tramite seriale.
 - **LoggerView:** rappresenta il componente utile a trascrivere aggiornamenti sullo stato del sistema e sulle scansioni.
- **Controller:** è il componente che, collegato a SmartRadarView, reagisce agli eventi inviati dall'utente e comunica con lo SmartRadar mediante la comunicazione seriale.