RELAZIONE PROGETTO#2 – Smart Experiment

È stato realizzato un sistema embedded che permette di fare esperimenti di cinematica, tracciando la posizione, la velocità e l'accelerazione di un corpo che si muove su una linea retta.
Il sistema proposto si compone di due parti: un sottosistema su Arduino realizzato con un approccio a task ed un sottosistema su PC che permette di tracciare i dati registrati durante l'esperimento.
Le due parti comunicano mediante scambio di messaggi sottoforma di stringhe comunicati tramite Seriale.

Il sottosistema Arduino è organizzato nei seguenti Task:

- ExperimentTask: Task periodico, con frequenza regolabile prima dell'inizio dell'esperimento tramite potenziometro. Questo task si occupa della completa gestione dell'esperimento durante i suoi diversi stati.
- DetectPresenceTask: Task di periodo 100 ms che definisce il comportamento del Pir, permettendo di sapere se è stata rilevata la presenza di qualcuno.
- GreenLedTask: Task di periodo 100 ms che definisce il comportamento del led L_1 durante le diverse fasi dell'esperimento.
- RedLedTask: Task di periodo 100 ms che definisce il comportamento del led L_2 durante le diverse fasi dell'esperimento.
- BlinkTask: Task di periodo 40ms che permette il lampeggiamento del Led L_2 quando necessario.
- StartButtonTask: Task di periodo 100 ms che definisce il comportamento del bottone B_{start}.
- StopButtonTask: Task di periodo 100 ms che definisce il comportamento del bottone B_{stop}.

Le interfacce Button.h, Light.h, Pir.h, ServoMotor.h, TemperatureSensor.h e Sonar.h rappresentano le astrazioni usate dai Task dei diversi sensori/attuatori realizzate con classi astratte.

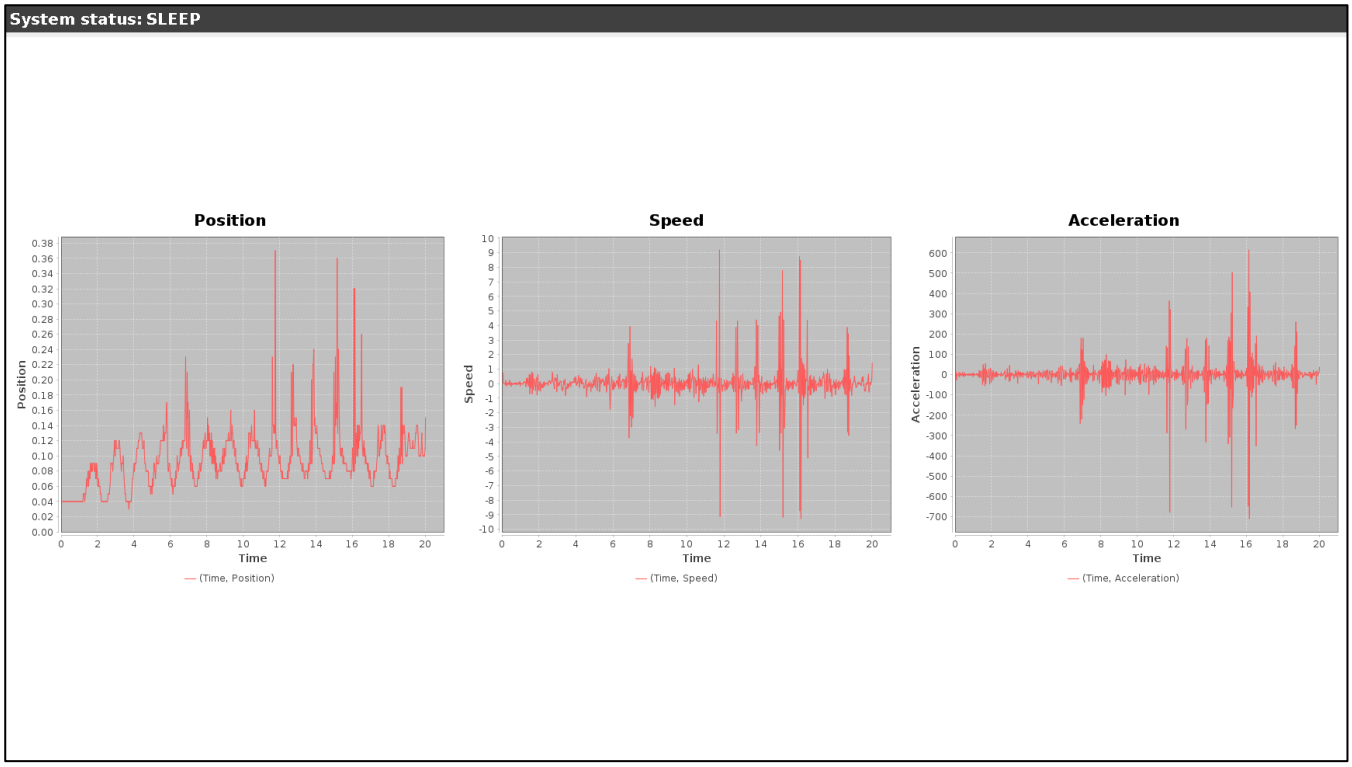
ButtonImpl.h, Led.h, PirImpl.h, ServoMotorImpl.h, TemperatureSensorImpl.h e SonarImpl.h rappresentano invece le dichiarazioni delle classe concrete.

Il comportamento dei Task è descritto più nello specifico nei diagrammi presenti nel file 'experiment.pdf'.

Il sottosistema Java su PC è invece composto come segue:

- SerialManager: rappresenta il model del sottosistema. In particolare, si occupa della ricezione e dell'interpretazione dei dati dalla Seriale e dell'invio al sottosistema Arduino della conferma utente a fine esperimento. Questo è reso possibile dal CommChannel, che si occupa nello specifico della comunicazione.
- Controller: rappresenta il Controller del sistema. Permette quindi la comunicazione dei dati tra SerialManager e GUI.
- Viewer: rappresenta l'interfaccia grafica del sottosistema. La rappresentazione dei grafici che modellano l'andamento nel tempo della posizione, della velocità e dell'accelerazione dell'oggetto è stata fatta usando la libreria JFreeChart.

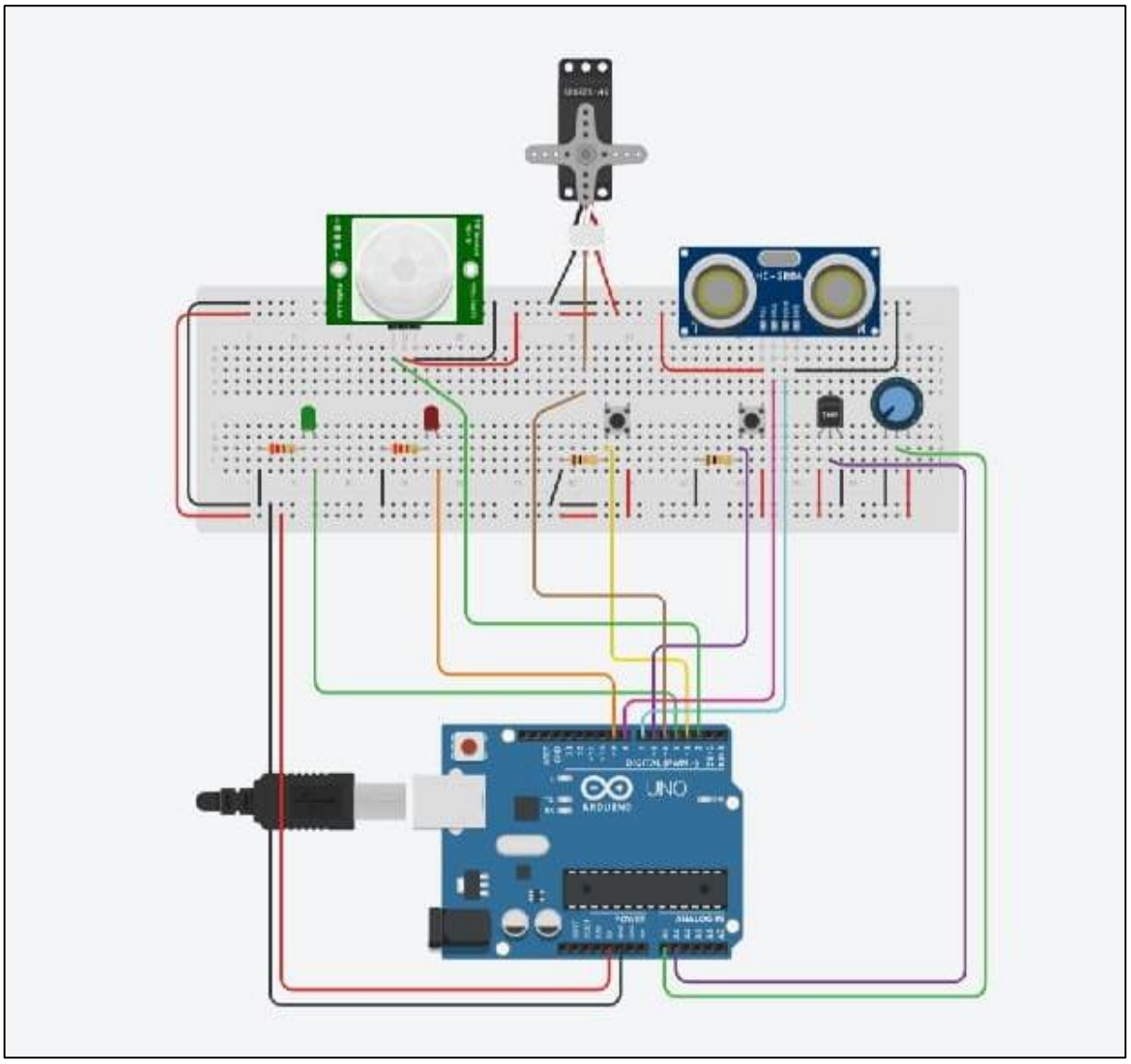
Di seguito uno screenshot dei grafici di un esperimento:



L'interfaccia grafica mostra i diversi stati dell'esperimento che possono essere:
- System status : waiting for connection
- System status : pir calibration
- System status : waiting
- System status : sleep
- System status : error
- System status : experiment started

Quando l’esperimento finisce appare invece una finestra pop-up nella quale si richiede la conferma da parte dell’utente.

Di seguito lo schema TinkerCad del circuito:



Progetto realizzato da:
Barzi Eddie
Belloni Sofia
Vissani Filippo