BEERBOT

“In the twenty-first century, robots will take the place which slave labor occupied in ancient civilization.” Nikola Tesla

Code Repository:
https://github.com/luca493/BeerBot

Robotics & Design **Politecnico di Milano**

Nicola Di Nardo
Marco Ballabio
Pedro Simon Sanchez
Tobias Pfhul
Luca Bossi

Index

Abstract

"The level of excitement and expectation reminds me so much of that time when Paul Allen and I looked at the convergence of new technologies and dreamed of the day when a computer would be on every desk and in every home... I can envision a future in which robotic devices will become a nearly ubiquitous part of our day-to-day lives."

Bill Gates

Within the report, the design and features of BeerBot robot will be described. BeerBot is an advertisement robot design for commercial uses. The aim of the robot, which stands on a table and has the appearance of a beer mug, would enhance the people to get into the public place it is promoting (pub, bar, club and so on).

First of all, the main state of the art will be presented, related to the inspiration, which had been taken to develop BeerBot. Secondly, a brief sketch of the robot will be presented, in which its main components will be shown. What is more, the explanation will go through all the sensors and actuators in order to provide a full description of the product.

However, the report does not only focus on the robot by itself, but also on the interactions between the robot and the environment (mainly supported by different kind of diagrams) in order to allow not expert people to understand the behavior of BeerBot. Lastly, some annexes are added in order to fully give an approach to the development of the robot.

Description

The aim of the course for the current edition is to design and prototype robots, i.e., devices with sensors and actuators able to perform autonomous activity, that could engage people in public spaces such as shop windows, malls, stations, museums, and attract their attention to something that deserves it, like goods to be advertised, security ads, special displays, etc. New ways to attract attention and engage people distracting them from their activities are needed, and movable objects with sensors, such as the ones to be developed here, are one of the most interesting mean.

Our group decided to build a robot with the appearance of a beer mug (hence the name "BeerBot") in order to attract people inside different types of public places, such as pubs, breweries, clubs and so on. The design objective we set up was to build a robot with a simple and clean look.

The aim of the product is to interact with passersby in a funny, interesting and autonomously way, in order to catch their attention, first on the robot and then on the club it is promoting. In fact, BeerBot is able to detect people close to him either with "sight" or with "touch" in order to make interaction with the client possible.
BeerBot will make many friends, which means that so many beers will be poured!

Research

State of the Art

Robotics devices for attrcating customers' attention and promote brands have been researched for a long time. We mainly focus our attention on the queries "robot" and "promotion".

1. *Pepper*

Brand: Parco 2

Market Field: Shopping Center

Location: Sendai

Description: The robot is used to attract clients and guide them to shops they are interestend in

Source: www.youtube.com/watch?v=rhuKQ32b5TM

2. *Promobot*

Brand: Promobot

Market Field: Any

Location: Any

Description: Promobot is a robot designed to attract and help customers. It is able to interact with clients and show them emotions

Source: www.en.promo-bot.ru

Concept

Inspiration

The main idea was to build a marketing-oriented robot capable to catch the attention of customers in public places. After several ideas, we decided to promote a specific kind of public places, such as pubs or breweries with a robot shaped as a beer mug. Our main inspiration for the prototype design were anthropomorphic objects, as they can be seen in many animation movies. For instance, a discarded idea was a robotic movable table, which is shown in one of the following pictures. Then, watching several videos on Youtube gave us the idea of giving personality to the robot and we began to think about an animated beer mug that became our final choice.

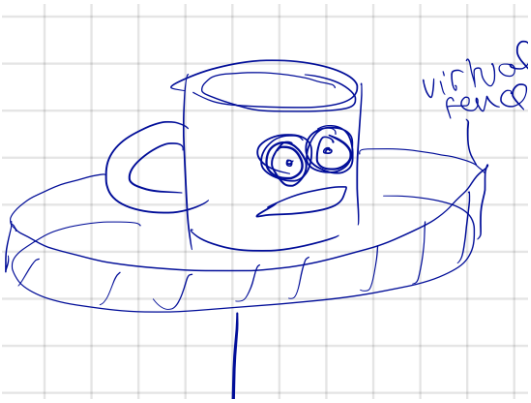
Robotic moving table

Dancing mustaches

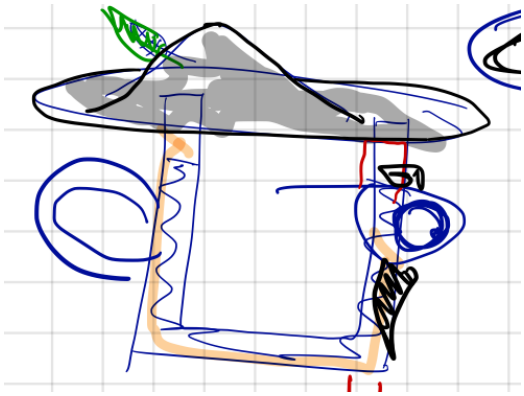
Representation

Concept "Art" of BeerBot:

- **Drawings**



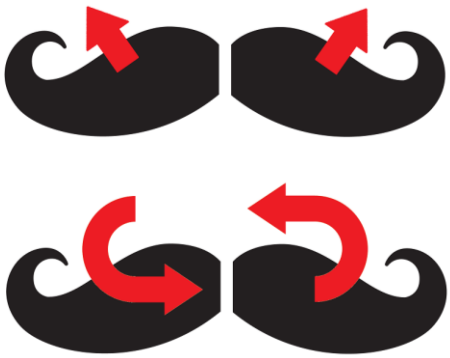
Our first idea was to use a table with a vertical fence in order to keep the robot stands on top of the table



Here the robot wears a traditional Bavarian hat, which will be discarded to be replaced by the foam

First model design

- **Design**



Designing the movement of mustaches

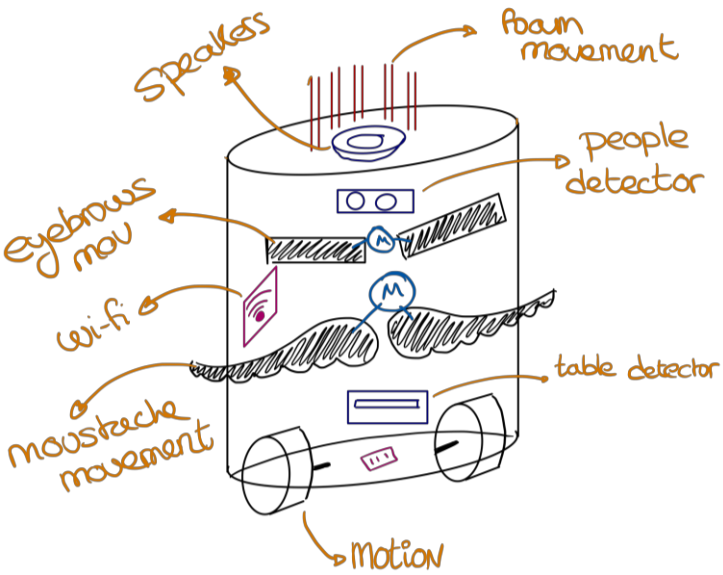
Two Servo motors that allow the movement of both mustaches and eyebrows

The mechanism used to move both mustaches and eyebrows was built using Lego.

Internal Structure

External Structure

At this stage both internal and external structure has been made using cardboard, but it will be replaced respectively by a wooden and a plastic structure for the final prototype.



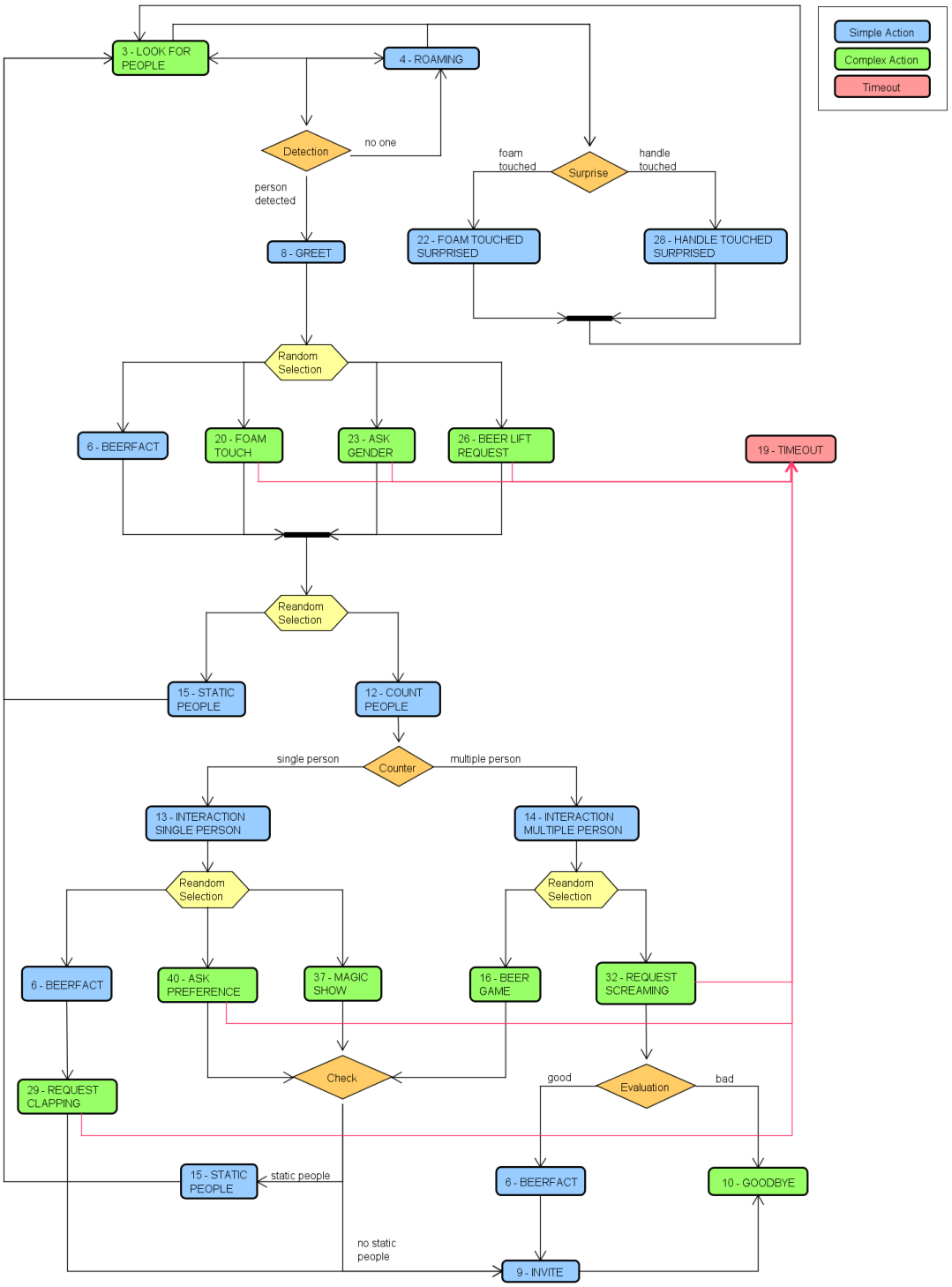
This is the model of the mid-term prototype. It has almost all the components and functionalities of the final robot. Notice that there is also a wi-fi module to allow remote control of the robot, an idea that has been left out for design reasons

Mid-term BeerBot

Development

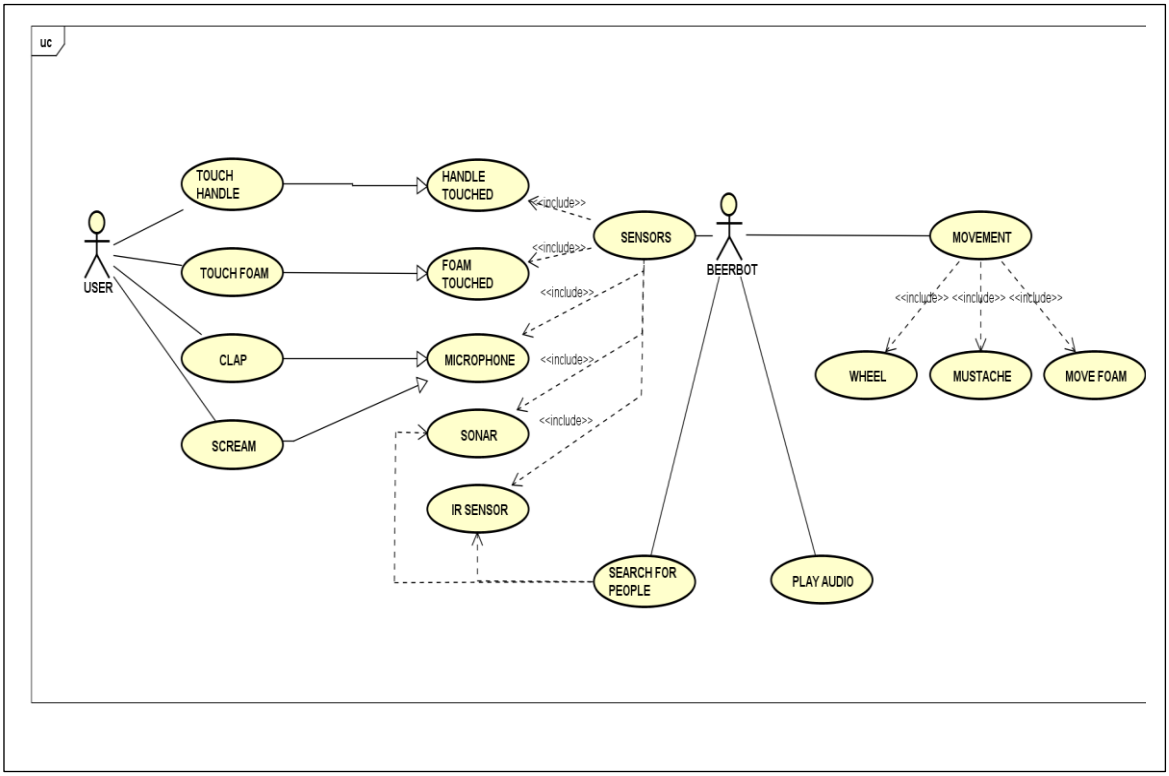
Interaction

Flowchart



We choose to show a Flowchart of the robot's behavior in order to give an overview of all its functionalities to users. The Flowchart is a diagram that shows the workflow of the process, showing all the steps as boxes of various types and their order by connecting them with arrows.

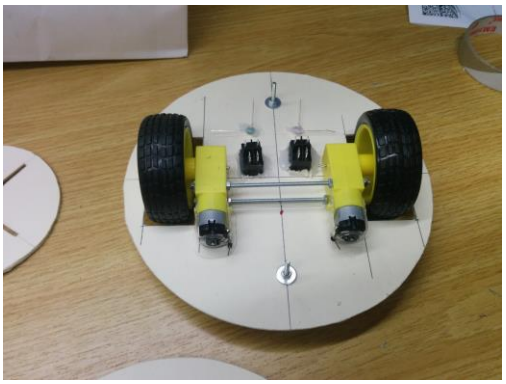
Use Case Diagram



The Use Case Diagram shows the simplest representation of a user's interaction with the system that exhibits the relationship between the user and the different use cases in which the user and system are involved. It provides a high-level overview of the system, a simplified and graphical representation of what the system actually does.

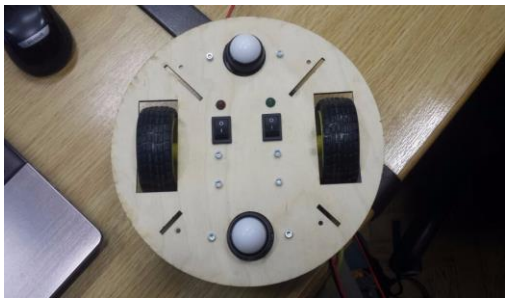
Shape

- **Base**



Top view

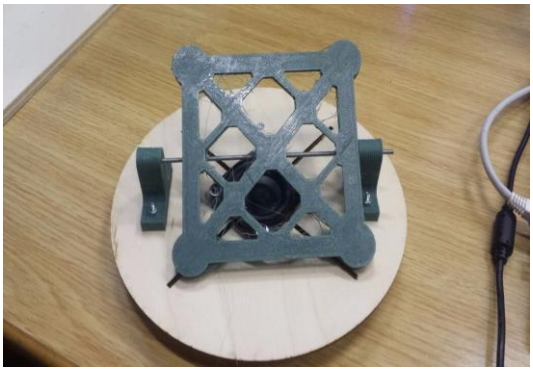
Round base with two wheels actuated by the two DC motors



Bottom view

Two ball caster wheels for stability reasons and two switches, one for mechanics and one for the electronics

- **Foam**



Top view

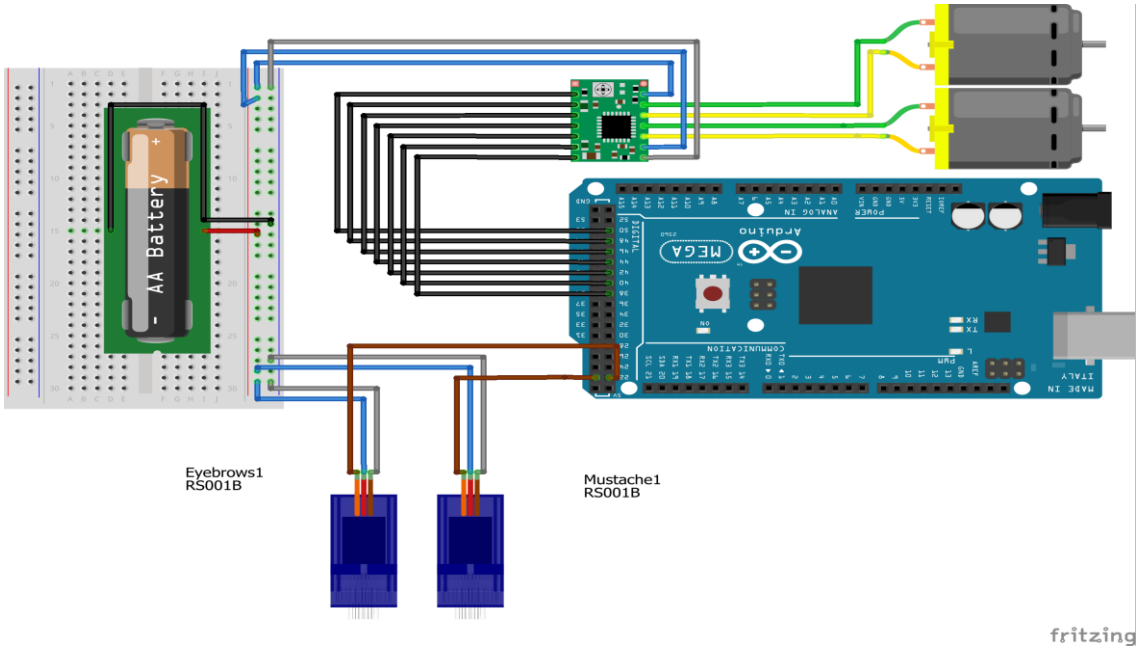
Mechanism to move and attach the foam

Foam to put on top of the robot

- **Internal Structure**
- **External Structure**

Mechanics

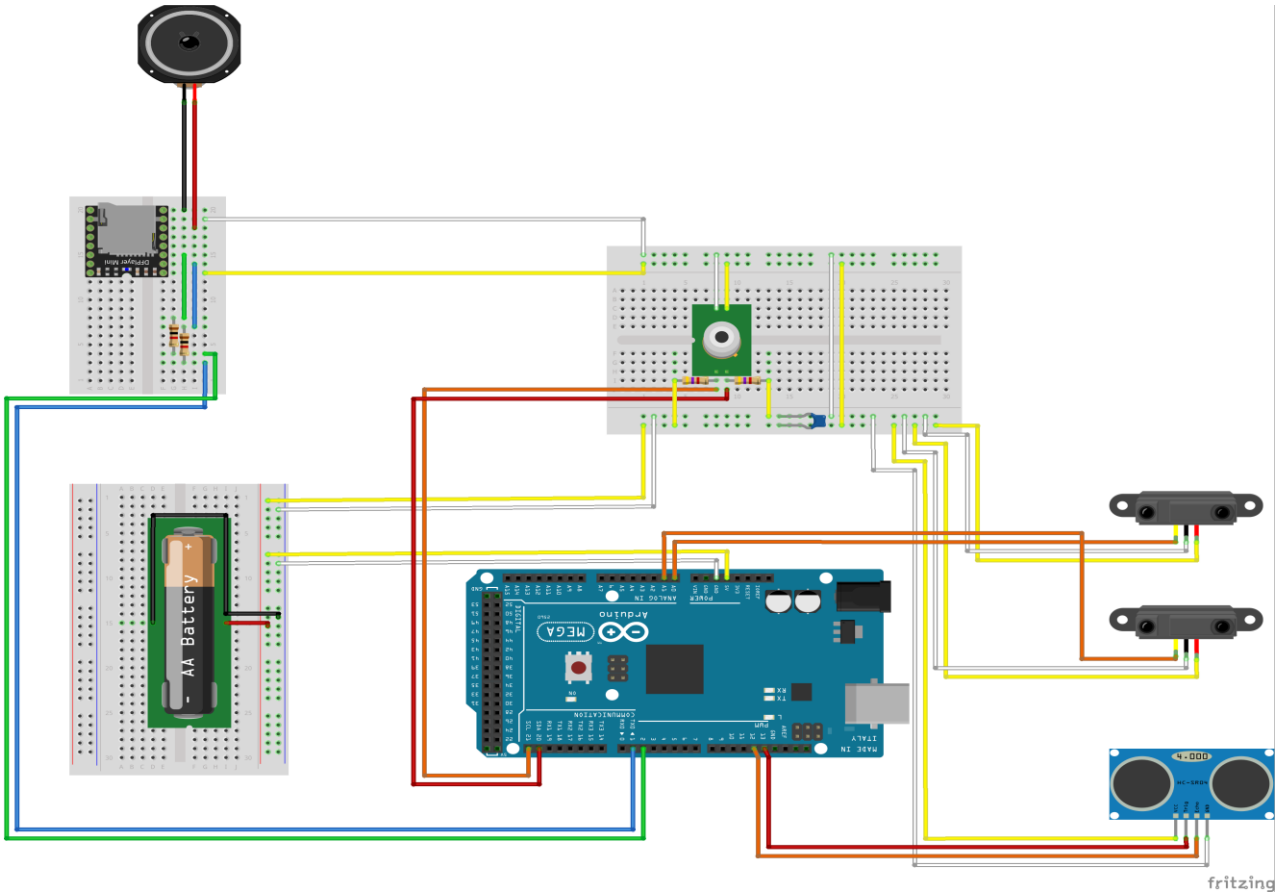
Fritzing circuit representation of mechanics used to move the robot.



Two DC motors on the top-left corner, controlled by the Pololu driver, used to move the robot on the table and the two Servo motors in the bottom part used to move mustaches and eyebrows.

Electronics

Fritzing circuit representation of electronics used to allow the robot to sense the environment, talk and stand on the table.

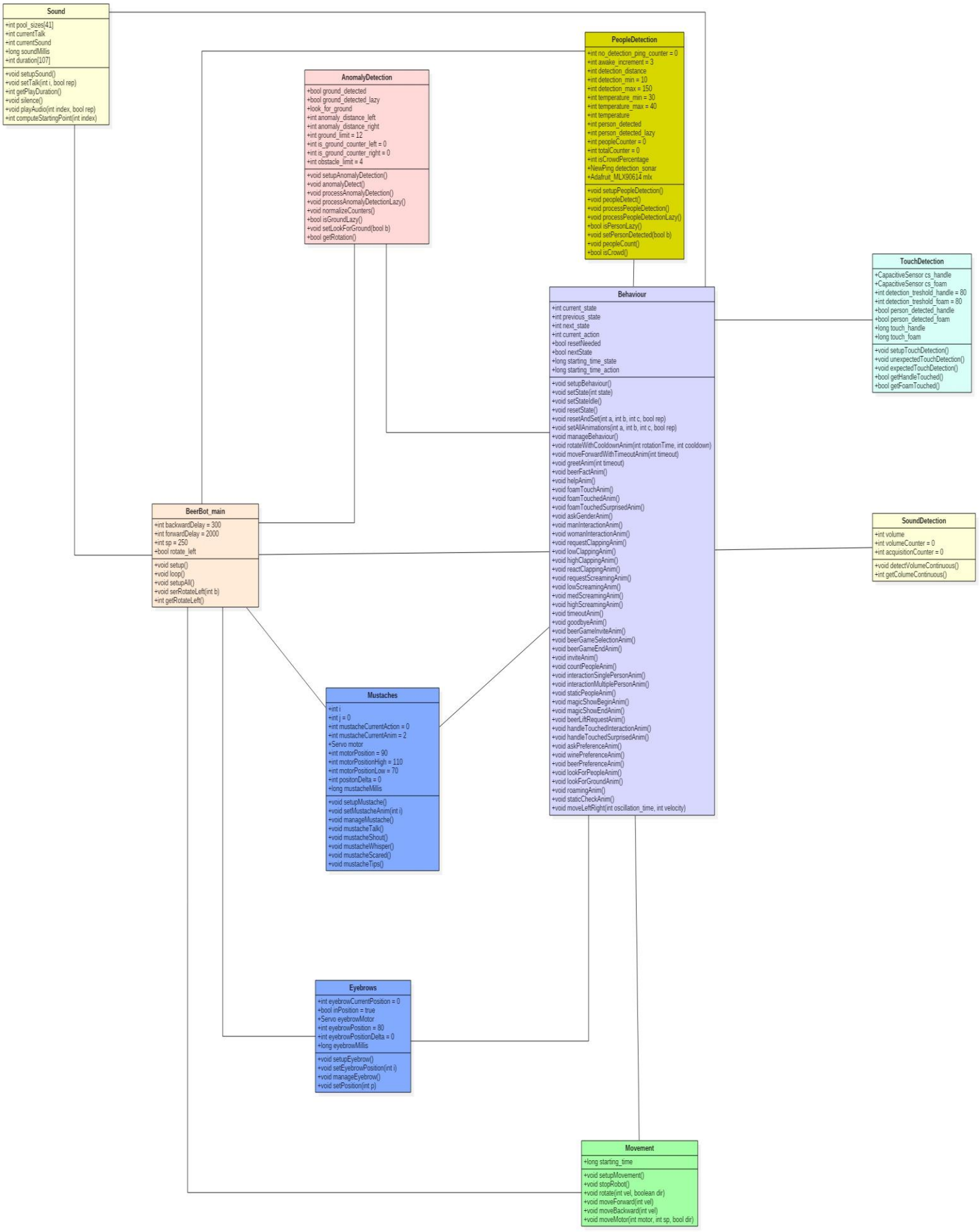


Two InfraRed sensors (black ones on the right) that allows the robot not to fall down, the Sonar sensor (bottom-right corner) and the InfraRed Thermo sensor (on top of the picture) which are used to detect people and the MiniDFPlayer linked to a speaker (top-left corner) that allows the robot to talk with users.

Informatics

In order to give an overview of the system at code-level the section shows a sort of class diagram. The term "class diagram" does not perfectly fit the actual system because there is no concept of class, in the strict sense, neither in the Arduino's programming language or in our implementation. However, the idea of a static representation of the system still works for our product, even if it is not an accurate one. The diagram describes the structure of the system by showing for each file their variables, methods and the relationships among them.

Shown in the following page.


Sound
+int pool_sizes[41]
+int currentTalk
+int currentSound
+long soundMillis
+int duration[107]
+void setupSound()
+void setTalk(int i, bool rep)
+int getPlayDuration()
+void silence()
+void playAudio(int index, bool rep)
+int computeStartingPoint(int index)
AnomalyDetection
+bool ground_detected
+bool ground_detected_lazy
+look_for_ground
+int anomaly_distance_left
+int anomaly_distance_right
+int ground_limit = 12
+int is_ground_counter_left = 0
+int is_ground_counter_right = 0
+int obstacle_limit = 4
+void setupAnomalyDetection()
+void anomalyDetect()
+void processAnomalyDetection()
+void processAnomalyDetectionLazy()
+void normalizeCounters()
+bool isGroundLazy()
+void setLookForGround(bool b)
+bool getRotation()
PeopleDetection
+int no_detection_ping_counter = 0
+int awake_increment = 3
+int detection_distance
+int detection_min = 10
+int detection_max = 150
+int temperature_min = 30
+int temperature_max = 40
+int temperature
+int person_detected
+int person_detected_lazy
+int peopleCounter = 0
+int totalCounter = 0
+int isCrowdPercentage
+NewPing detection_sonar
+Adafruit_MLX90614 mlx
+void setupPeopleDetection()
+void peopleDetect()
+void processPeopleDetection()
+void processPeopleDetectionLazy()
+bool isPersonLazy()
+void setPersonDetected(bool b)
+void peopleCount()
+bool isCrowd()
TouchDetection
+CapacitiveSensor cs_handle
+CapacitiveSensor cs_foam
+int detection_treshold_handle = 80
+int detection_treshold_foam = 80
+bool person_detected_handle
+bool person_detected_foam
+long touch_handle
+long touch_foam
+void setupTouchDetection()
+void unexpectedTouchDetection()
+void expectedTouchDetection()
+bool getHandleTouched()
+bool getFoamTouched()
Behaviour
+int current_state
+int previous_state
+int next_state
+int current_action
+bool resetNeeded
+bool nextState
+long starting_time_state
+long starting_time_action
+void setupBehaviour()
+void setState(int state)
+void setStateIdle()
+void resetState()
+void resetAndSet(int a, int b, int c, bool rep)
+void setAllAnimations(int a, int b, int c, bool rep)
+void manageBehaviour()
+void rotateWithCooldownAnim(int rotationTime, int cooldown)
+void moveForwardWithTimeoutAnim(int timeout)
+void greetAnim(int timeout)
+void beerFactAnim()
+void helpAnim()
+void foamTouchAnim()
+void foamTouchedAnim()
+void foamTouchedSurprisedAnim()
+void askGenderAnim()
+void manInteractionAnim()
+void womanInteractionAnim()
+void requestClappingAnim()
+void lowClappingAnim()
+void highClappingAnim()
+void reactClappingAnim()
+void requestScreamingAnim()
+void lowScreamingAnim()
+void medScreamingAnim()
+void highScreamingAnim()
+void timeoutAnim()
+void goodbyeAnim()
+void beerGameInviteAnim()
+void beerGameSelectionAnim()
+void beerGameEndAnim()
+void inviteAnim()
+void countPeopleAnim()
+void interactionSinglePersonAnim()
+void interactionMultiplePersonAnim()
+void staticPeopleAnim()
+void magicShowBeginAnim()
+void magicShowEndAnim()
+void beerLiftRequestAnim()
+void handleTouchedInteractionAnim()
+void handleTouchedSurprisedAnim()
+void askPreferenceAnim()
+void winePreferenceAnim()
+void beerPreferenceAnim()
+void lookForPeopleAnim()
+void lookForGroundAnim()
+void roamingAnim()
+void staticCheckAnim()
+void moveLeftRight(int oscillation_time, int velocity)
BeerBot_main
+int backwardDelay = 300
+int forwardDelay = 2000
+int sp = 250
+bool rotate_left
+void setup()
+void loop()
+void setupAll()
+void serRotateLeft(int b)
+int getRotateLeft()
SoundDetection
+int volume
+int volumeCounter = 0
+int acquisitionCounter = 0
+void detectVolumeContinuous()
+int getColumeContinuous()
Mustaches
+int i
+int j = 0
+int mustacheCurrentAction = 0
+int mustacheCurrentAnim = 2
+Servo motor
+int motorPosition = 90
+int motorPositionHigh = 110
+int motorPositionLow = 70
+int positonDelta = 0
+long mustacheMillis
+void setupMustache()
+void setMustacheAnim(int i)
+void manageMustache()
+void mustacheTalk()
+void mustacheShout()
+void mustacheWhisper()
+void mustacheScared()
+void mustacheTips()
Eyebrows
+int eyebrowCurrentPosition = 0
+bool inPosition = true
+Servo eyebrowMotor
+int eyebrowPosition = 80
+int eyebrowPositionDelta = 0
+long eyebrowMillis
+void setupEyebrow()
+void setEyebrowPosition(int i)
+void manageEyebrow()
+void setPosition(int p)
Movement
+long starting_time
+void setupMovement()
+void stopRobot()
+void rotate(int vel, boolean dir)
+void moveForward(int vel)
+void moveBackward(int vel)
+void moveMotor(int motor, int sp, bool dir)

Conclusions

Working on this prototype we learned various things:

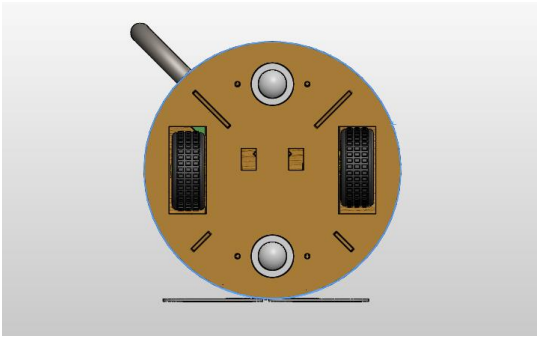
- The importance of having different components and a modular structure in order to allow fast prototyping.
- The importance of organization while working in team.
- The amount of programming errors that can appear while coding on Arduino and the inadequacy of the Arduino IDE.
- The difficulty of assembly all the connections and cables inside a structure.
- The difficulty to design the behavior of the robot in order to meet both requirements and our objectives.
- The complexity in designing the assembly so it can be unassembled easily to
    - Charge the battery
    - Reconnect cables
    - Add devices
    - Maintenance
    - Implement extensions

Annexes

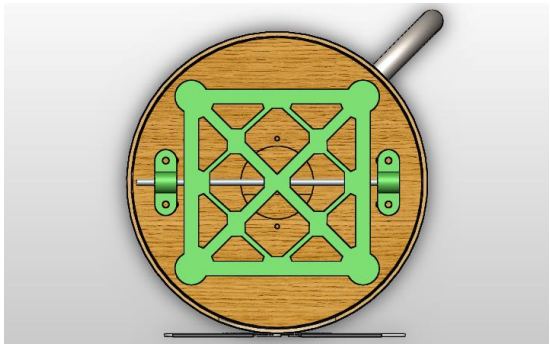
Shape - 3D Model

Now we want to shows our 3D models from all point of views in order to give a reference with respect to the actual robot.

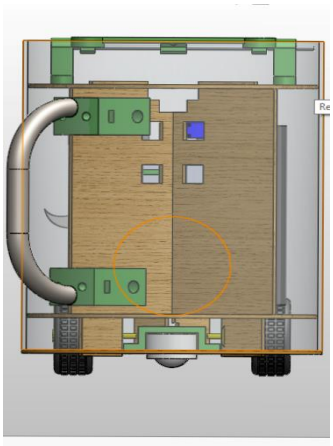
- **Bottom view**



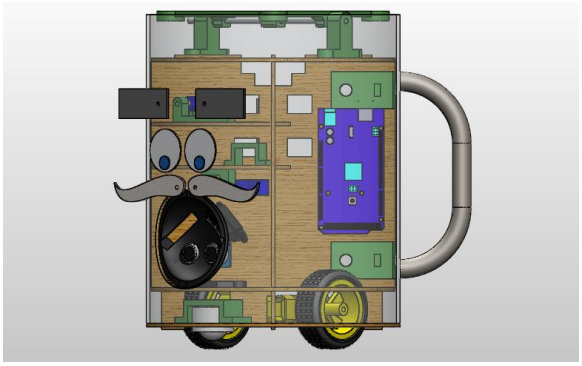
- **Top view**



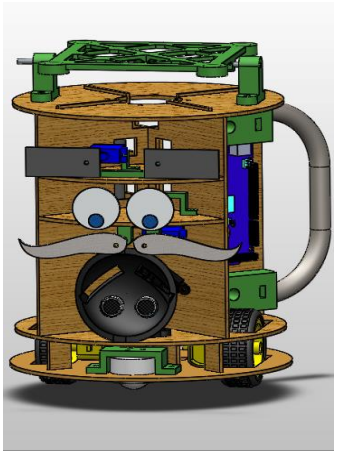
- **Back view**



- **Side view**



- **Front view**



Final shape

Electronic Devices

List of components of the robot:

- **2 Servo Motors**

- **2 Infra Red Proximity Sensors**

- **2 DC Motors**

- **1 MiniDFPlayer**

- **1 Speaker**

- **1 Sonar Sensor**

- **1 Pololu Driver**
  (tb6612fng)

- **1 Gaoxing Tech Microphone**
  (TS-GX-115)

- **1 Infra Red Thermometer**
  (MLX90614)

- **2 Capacitive Sensor**

- **1 Turnigy 5.0 Battery**

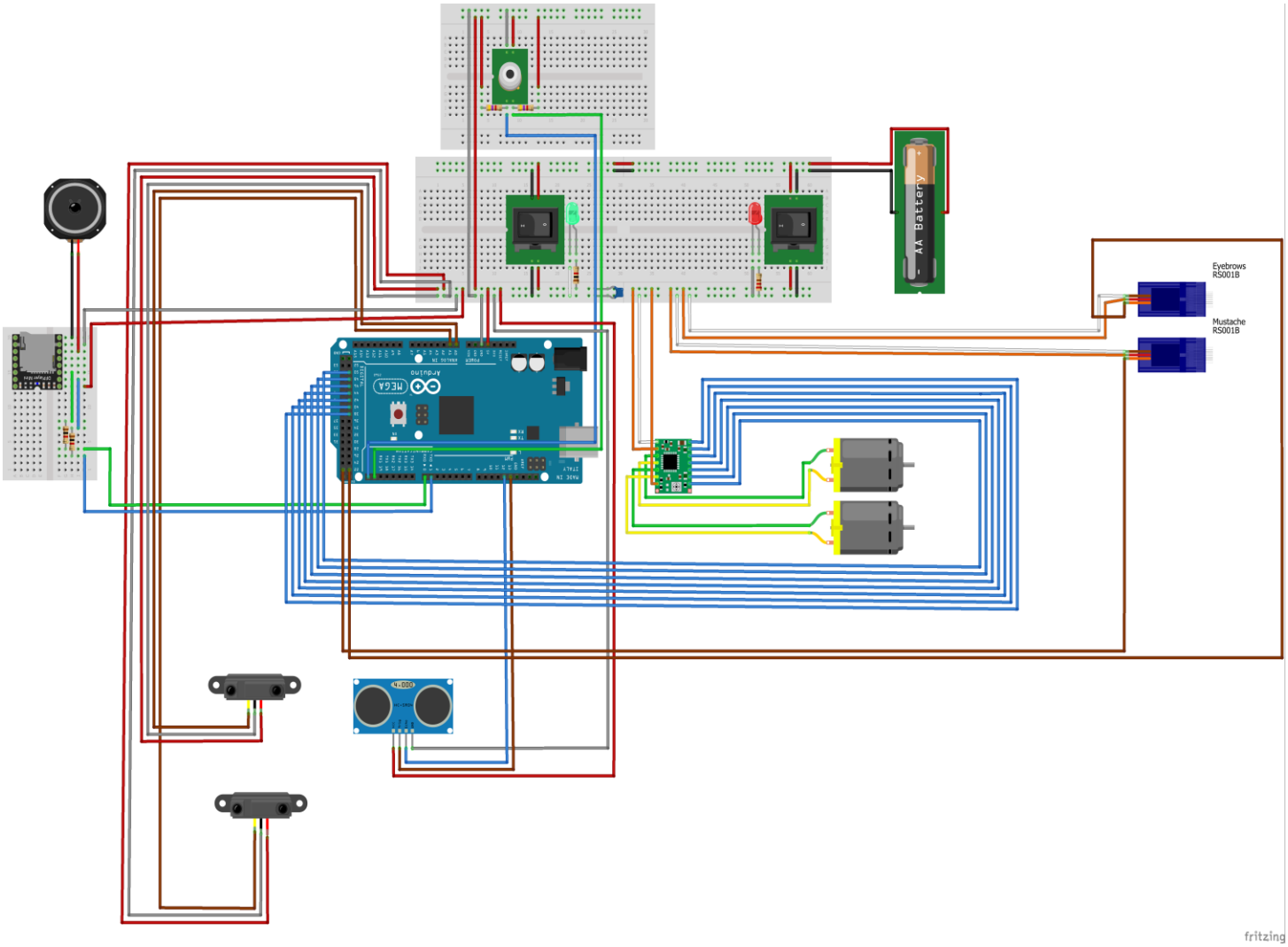
- **1 Ubec Buzzer 3A**

- **2 LED**

- **Arduino Mega2560**

Complete Circuit – Fritzing

The following diagram shows the entire circuit of the system, with all the components (sensor, actuators, Arduino board). For detailed information about each single component go to *Electronic Devices* section.



MLX90614ESF-ASS-000-SP Circuit

(Temperature Sensor)

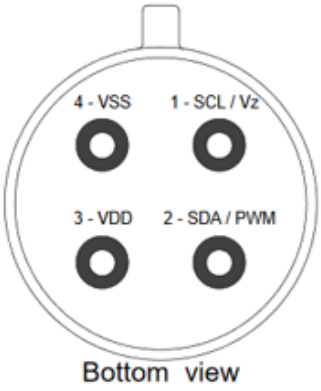
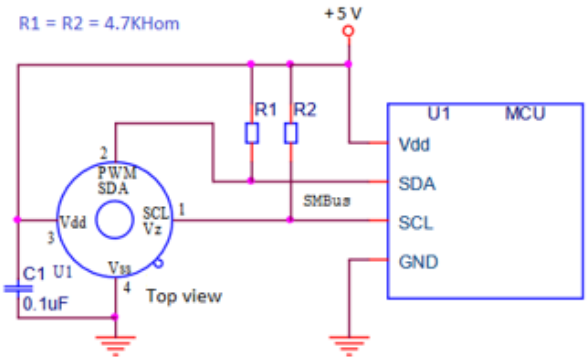


Figure 2: Pin description



Library:

https://github.com/sparkfun/SparkFun_MLX90614_Arduino_Library