# SIMBA: a Sound localisation assistance module

Text

Description automatically generated

## **System overview**

**Simba** is an add-on to the **Rover V2** Arduino-compatible robot. It uses a colour sensor and three microphones with integrated noise filters, to locate and navigate towards a sound source. On detecting a red light, it stops.

**Unique features**

* Gradual deceleration before stopping 1 cm from the target
* 360° detection of sound
* Filters out sharp obscure noises like footsteps, distant conversation, etc.

**Optimal operational conditions:**

* Rigid surface or level ground
* Silent indoor environment
* Fully charged batteries
* Within 1.1 metres of the sound source

**Adafruit TCS34725 Light & Colour Sensor:**

The Light and Colour Sensor is mounted at the front of the robot and is used to obtain a measurement of the *colour temperature (K)*. The analogue measurement from the sensor determines the speed of the Rover V2’s motors, allowing it to slow to a halt in front of the target.

**Required installed software packages:**

* Adafruit TCS34725
* Adafruit BusIO

**Grove Seeed Studio LM358 Amplified Sound Sensor:**

The 3 Seeed Studio microphones are mounted on the left, right and rear of the Rover V2. The left and right microphones are used determine the angle to the target, while the rear microphone determines whether the target is in front of or behind the Rover V2.

The signal from the microphones is filtered through three deparated integrated passive high pass filter. The microphone array does not require any additional software packages to be installed.

**Data path overview:**

Diagram

Description automatically generated

This diagram shows the data path of two of the major sub-systems, the colour detection, and the sound detection.

After an initial measurement of the rear microphone to determine whether the target is in front of or behind the robot, the ATMEGA328P microcontroller (within the Arduino Uno) uses *polling* to repeatedly measure the analogue signals from the left and right microphones.

The processor ‘simultaneously’ polls the *colour temperature* measurement from the Adafruit TCS34725 Light and Colour Sensor.

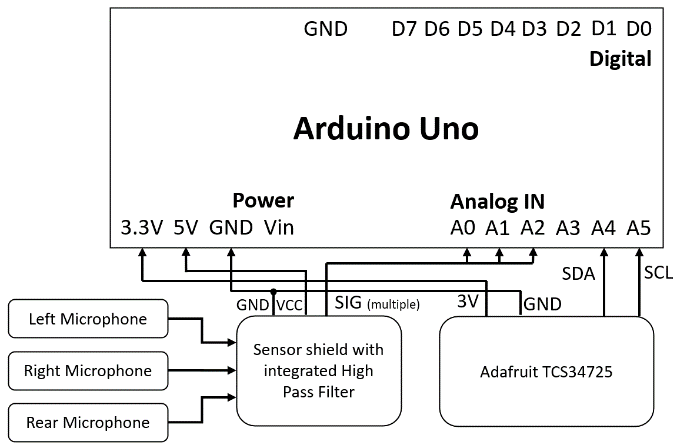
**Overview of shared power lines:**

Diagram

Description automatically generated

Every sensor in the system shares a common ground on the Arduino Uno. The three microphones are connected to the 5V pin, whilst the Adafruit TCS34725 Light & Colour Sensor is powered by the 3.3V pin.

## **Circuit Schematic**



Each microphone is connected to a passive high-pass filter which sends the filtered output to be read by an analogue pin.

|  |  |  |
| --- | --- | --- |
| **Microphone** | **Mic. Pin** | **Rover V2** |
| Left | SIG | A0 |
| VCC | 5V |
| GND | GND |
| Right | SIG | A1 |
| VCC | 5V |
| GND | GND |
| Rear | SIG | A2 |
| VCC | 5V |
| GND | GND |

The Adafruit TCS34725 Light and Colour Sensor uses I2C to interface with the Rover V2.

## **Mechanical construction**

The adapter (figure 1) is used to attach the caster wheel included to the chassis of the robot. The bottom four holes hold the caster wheel, and the top four holes connect to the body of the robot.



*Figure 1: Caster wheel adapter*

The microphone bracket (figure 2) fits on top of the robot and has six different positions (four at the front and two at the back) that each microphone can be fitted. It is recommended to use the two positions at the front which each face 45° outwards for the front-facing microphones and the vertical position at the back for the rear-facing microphones for the best results.

Diagram, engineering drawing

Description automatically generated

*Figure 2: Microphone bracket*

## **Software**

Diagram

Description automatically generated**Sound Source Localisation:**

***orientation***The robot uses 3 microphones to take short samples of the sound, before comparing the magnitude of the samples. Based on that, the robot orients itself to face the sound source.

***soundSampling***2 microphones at the front of the robot are simultaneously sampled, taking up 1200 bytes of available program memory.

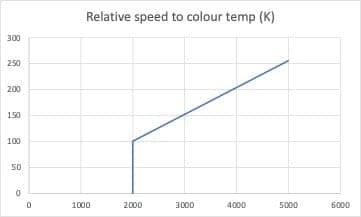
***soundSourceLocalisation***Subroutine used to determine the direction of the incoming sound. Algorithm calculates the difference in the time of arrival of the sound to each microphone.

Diagram, schematic

Description automatically generated

**Movement and stoppage:**A series of simple movement subroutines (forward, left, right) are responsible for guiding robot towards the sound source. The colour temperature readings from the colour sensor are used to scale the motor’s speed. The red LED that stops the robot has a colour temperature of 2000K and the colour sensor outputs 5000K when detecting bright sunlight, and the speed of the motor is adjusted proportionally by use of the map function. The temperature reading rarely rises above 5000K but, nevertheless, a constrain function has been implemented in the code to prevent the speed from rising above the maximum speed or falling below the minimum speed (255 and 0 respectively).

The graph below illustrates how the motor speed varies respective to the colour temperature reading. The motors destabilize when run at a relative speed below 100 and stop running altogether at a speed below 60 so 100 has been chosen as the minimum speed for the motors.



## **Test data**

The following graph illustrates how the Adafruit TCS34725 Light and Colour Sensor responds to a change in distance between the robot and the red LED, assuming the LED is directly in front of the sensor.

|  |  |  |
| --- | --- | --- |
| **TEST DATA** | | |
| Battery Life (Continuous Operation) | | 120 minutes |
| Charging Time | | 90 minutes |
| Battery Specifications | | 4 x Ni-Mh 2300mAh (rechargeable) |
| Nominal Power Consumption | | 1.34 mW |
| Nominal Voltage | | 5.0 V |
| Operational Voltage | | 4.8 to 5.3V |
| Microphone Array | Interface | TTL |
|  | Angular Range | 180 ± 5° |
|  | Sampling Frequency | 50 kHz |
| Colour Sensor | Interface | I²C |
|  | Angular Range | 70 ± 5° |
| Robot Speed | Linear | 91 mm/s |
|  | Angular | 72˚/s |
| Minimum Stoppage Distance | | 15 ± 5 mm |