The Running Alarm Clock

By Valerija Holomjova and Deborah Vella

***Abstract* - This report describes the robotic system designed and built for the Robotics Group Assignment. The robot that was designed is a running alarm clock that contains basic alarm functions and runs away from the user when an alarm goes off. The robot was given an obstacle avoidance feature so that it would not get stuck and prevent further movement. The aim of this project was to create a robot which forces the user to wake up and get out of bed to switch it off.**

I. DESIGNING THE ROBOT'S BODY

This section describes the two different designs we considered for our robot. Once a design was chosen, it was modelled using SketchUp [1].

A. First Design

The first design of the clock was different from the design that was decided to be implemented. The first design consisted of a cylindrical horizontal body, which would have contained all the circuits and electrical components needed. Please refer to Figure 1.1 for a sketch of this design. The following design had two wheels on either side of the body, which would have been bigger in diameter than the body to prevent the cylindrical section from reaching the floor in case the robot fell off the platform. The challenge of implementing this circuit would be trying to fit all the necessary electrical components inside a small space.

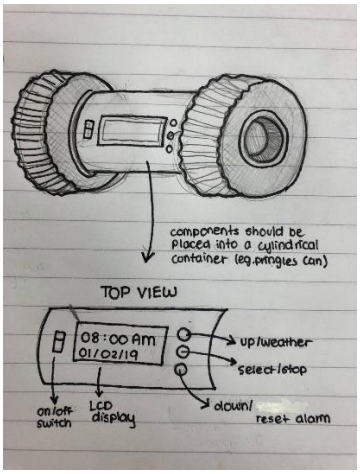

Figure 1.1 - A sketch of the first design.

B. Second Design

The second design consists of two main platforms that hold the circuits and all components used. This design uses the hardware components that were made available in the PiCar-S kit by SunFounder [2]. It was then altered by adding a second platform from another kit. Please refer to Figure 1.2 for a model representation of this design that was made in SketchUp. Please note that more screenshots of the model can be found in the Appendix of the report.

The following design uses four wheels instead of two, having the two rear wheels controlled by a DC motor each, while the front wheels are controlled by a servo motor. Therefore, the back wheels are used to move the robot forward or backwards, and the front wheels are used to change the direction of movement of the robot to the left or the right by rotating the servo motor. This design was chosen due to the time limit we had to build and implement the whole robot and due to our lack of hardware experience.

Components list for the robot's body:

- M2x8 Screws
- M2.5x6 Screws
- M3x8 Screws
- M3x8 Countersunk Screws
- M3x10 Screws
- M3x30 Screws
- M4x25 Screws
- M2 Nuts
- M2.5 Nuts
- M3 Nuts
- M4 Self-Locking Nuts
- M2.5x8 Copper Standoffs
- M3x25 Copper Standoffs
- 4x11x4 F694ZZ Flange Bearing
- Battery Holder
- DC Gear Motor
- SunFounder SF0180 Servo
- Rear wheels
- Front Wheels

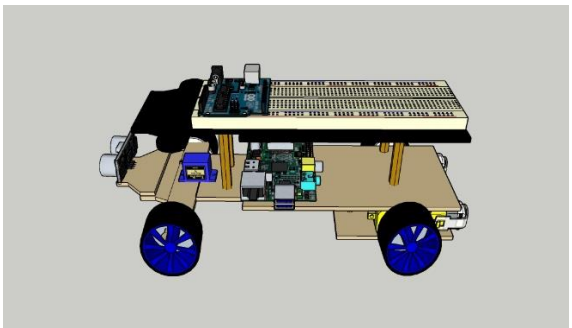

Figure 1.2 - A model of the second design on SketchUp.

II. DESIGNING THE CIRCUITS

Two main circuits were constructed - one circuit was controlled by Arduino and the other was controlled by the Raspberry Pi. The Arduino's circuit handles the alarm clock's functions, such as displaying the correct date and time, setting an alarm time, outputting a ringing tone when alarm goes off and more. On the other hand, the Raspberry

Pi's circuit controls the DC and Servo motors, together with the ultrasonic sensor placed at the very front of the robot. The Arduino and Raspberry Pi were connected with each other by a USB cable to be able to send signals from one to another through the serial terminal when the alarm is ringing or stopped. The Raspberry Pi circuit was implemented using the PiCar-S kit manual which came with the robot kit, however the Arduino circuit was design and implemented completely from scratch using prototyping, Circuito.io [3] and TinkerCad[4].

A. Arduino Circuit

The components required to build this circuit are:

- **Arduino** - used to upload code and give power to the circuit
- **Breadboard** - to connect the components with each other
- **LCD screen** - to display date and time, alarm time if any and displays what time the user is choosing when setting the alarm
- **Buzzer** - to make a sound when alarm goes off
- **Real Time Clock** - to keep track of the current date and time
- **Buttons** - to stop the alarm sound and movement, also used to navigate through the alarm functions such as setting the alarm time
- **Resistors (10k and 1k)** - used for the LCD and the buttons
- **Jumper wires** - to connect components
- **USB cable** - to connect the Arduino to Raspberry Pi to send signals to it

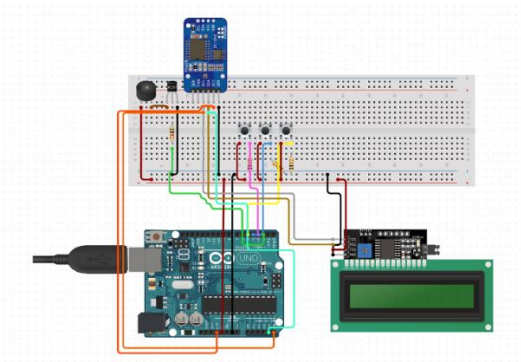

Figure 2.1 - The Arduino circuit of the robot.

B. Raspberry Pi Circuit

The components required to build this circuit are:

- Servo Motor
- Ultrasonic sensor
- Robot Hats
- Left and Right Motors
- PCA9685 PWM Driver
- TB6612 Motor Driver
- Jumper wires

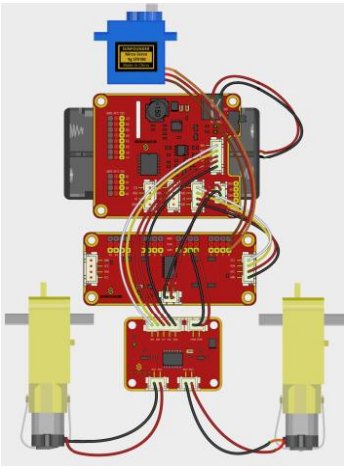

Figure 2.2 - The Raspberry Pi circuit of the robot.

III. CHALLENGES

One of the main challenges was building the alarm clock's body. The body was built, dismantled and rebuilt a couple of times, until the right design was achieved. Another challenge was to send signals from the Arduino to the Raspberry Pi at the correct time without any noticeable delays. This is because, at first, the Arduino code was sending a signal to the Raspberry Pi to start the movement routine and it worked as expected. On the other hand, when a button was pressed to stop the alarm, the sound from the Arduino was stopping immediately but the movement routine carried out by Raspberry Pi was not terminating. This issue was fixed through the code.

Another challenge was to make it as user friendly as possible, and at the same time it reaches its purpose. We considered that when an alarm goes off, the first human instinct might be to push the button and it forces the alarm to stop. To reduce this possibility, a small button was installed in the circuit, making it harder for the user to find it with closed eyes, and as a result, the clock gets enough time to start the movement routine.

Last but not least, the speed of the robot could also be considered as a challenge. The clock should not be running at a speed that makes it impossible for the human to catch, but at the same time it should move at a velocity which gives it a head start from the user. This was achieved by experimenting with the velocity and voltage passed to the motors.

IV. INTELLIGENCE

A. Path Finding

Having a running alarm clock, there will be the possibility of bumping into different obstacles found in its way. Hence, an ultrasonic sensor was used to calculate the distance between the robot's front and the obstacles in the way so that they could be avoided. Based on the read distance, the clock either continues moving forward (in the case of no obstacles), or changes direction to move left or right from the obstacle ahead, or reverses and turns if the obstacle is too close.

B. Google Assistant

Given more time, another type of intelligent system would have been implemented as an extra feature to the clock. The idea was to have the robot take in verbal instructions from the user and it would either speak back or display the requested information on the LCD. Such commands would have been a request for the current time, being able to set an alarm verbally, asking what the weather is for the day and checking what alarms have already been set. To implement this, we would have needed the google assistant application integrated with the Raspberry Pi, as well as a microphone and a speaker.

V. RESULTS AND CONCLUSION

Overall, we are satisfied with the outcome of the robot. The robot was able to interact successfully between the Arduino and the Raspberry Pi. We managed to implement basic alarm clock functions successfully such as setting alarms, resetting alarms, displaying the time and date, showing currently set alarms. The movement of the robot was also successful, the robot managed to dodge most obstacles and would start and stop moving parallel to the alarm ringing function. Future improvements include implementing a google assistant so that the user could use voice commands to interact with the robot. Furthermore, if this feature is implemented, voice commands could be used to check whether the user is awake by asking them a series of questions. Another improvement would be to create a more original and appropriate framework for the robot, that would help the robot to carry out its purpose in a more infallible way. This robot could also be taken further to act as a companion and perhaps have more functionalities implemented such as following a person, keeping track of important dates, playing music and setting reminders.

REFERENCES

[1] "3D Design Software | 3D Modeling on the Web," SketchUp. [Online]. Available: https://www.sketchup.com/. [Accessed: 28-May-2019].

[2] "SunFounder PiCar-S Kit V2.0 for Raspberry Pi 3/2/B ," Mazentop Demo. [Online]. Available: https://www.sunfounder.com/picar-s-kit.html. [Accessed: 28-May-2019].

[3] "Circuit Design App for Makers- circuito.io," *Circuit Design App for Makers- circuito.io*. [Online]. Available: https://www.circuito.io/. [Accessed: 28-May-2019].

[4] "From mind to design in minutes," Tinkercad. [Online]. Available: https://www.tinkercad.com/. [Accessed: 28-May-2019].

APPENDIX

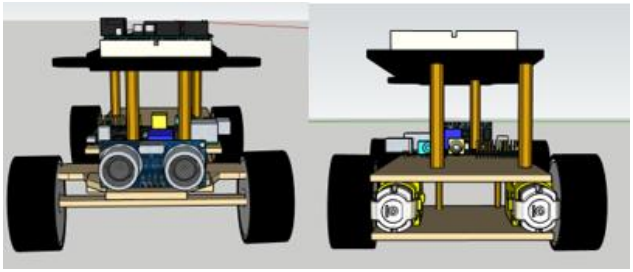

Figure 3.1 – Front and rear view of robot model

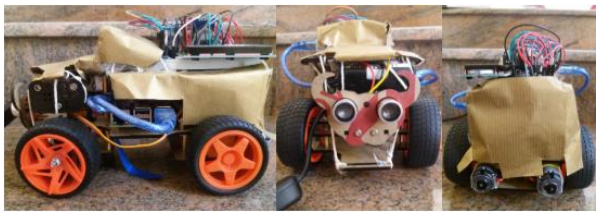

Figure 3.3 - Different angle views of the actual built robot. (it is wrapped in paper to cover the interior wires and important components for safety)

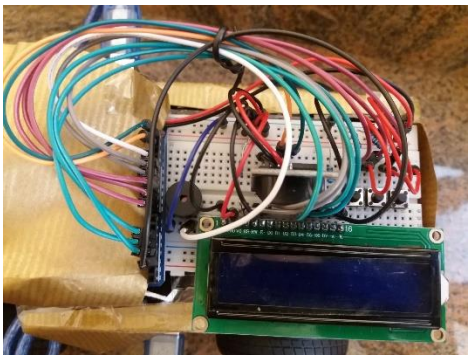

Figure 3.4 - Implemented Arduino circuit mounted on robot.

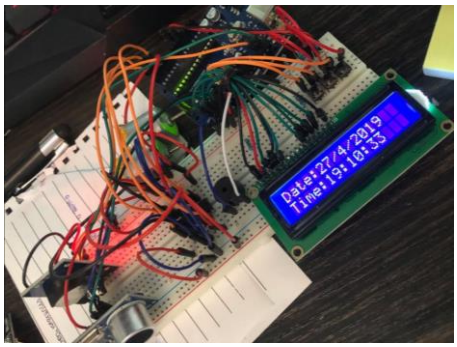

Figure 3.5 - Implemented Arduino circuit showing date and time.

Figure 3.5 – Building process of the robot.