CG1112 Engineering Principle and Practice

Semester 2 2018/2019

"Alex to the Rescue"

Final Report

Team: 02-05-01

Name	Student #	Main Role
Chua Zong Wei	A0183028H	Software
Chua Mai Feng Carl	A0182772X	Software
Lee Zhen Yu	A0182716B	Software
Edmund Teo Wei Xin	A0189971H	Hardware
Mohamed Nizar	A0183398M	Hardware

Section 1 Introduction (All)

72 hours. That is the golden period to locate and rescue survivors in the aftermath of disasters like earthquake, landslides or terrorist attacks. Against the ticking clock, rescuers have to risk their life to navigate through harsh environments like narrow passages, debris or even hazardous conditions to look for signs of life. In fact, there are approximately 100 firemen deaths in the US each year. In the process of rescuing another a life lost, defeats its purpose and is not exactly the ideal situation. Fortunately, recent robotic advancements have crafted many new possibilities for these rescue teams to utilize. I believe our team have come up with a robot that can contribute greatly to reducing the number of firemen deaths and allowing rescue teams to carry out rescue missions in a safer and more efficient manner.

Section 2 Review of State of the Art

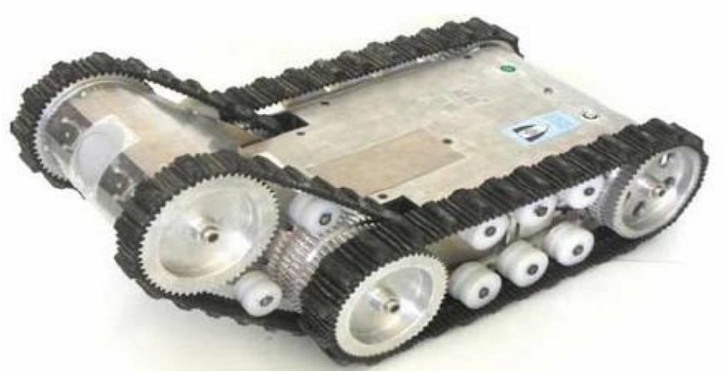

Figure 1. RAPOSA robot

RAPOSA:

- Designed to operate in disaster areas or hazardous environments
- Equipped with tracked wheels that allows locomotion on both sides, allowing operation in tough terrains
- The front body has one thermal camera to detect human heat signatures and two web-cameras to update the user on its surroundings. Additional features include gas, temperature and humidity sensors, light diodes, microphone and loudspeaker.
- This robot uses wireless communications, with an option for tethered operation
- Strengths:
    - When the robot is flipped over, it is still able to move and will flip its camera to compensate for the change
    - The front body of the robot can be adjusted to allow easier traversal across uneven ground
- Weaknesses:
    - Wireless communications may not work well due to electromagnetic interference and will require a tether that may restrict movement

    - Installing a tether will cause drag to the robot's movement and introduce the possibility of getting trapped

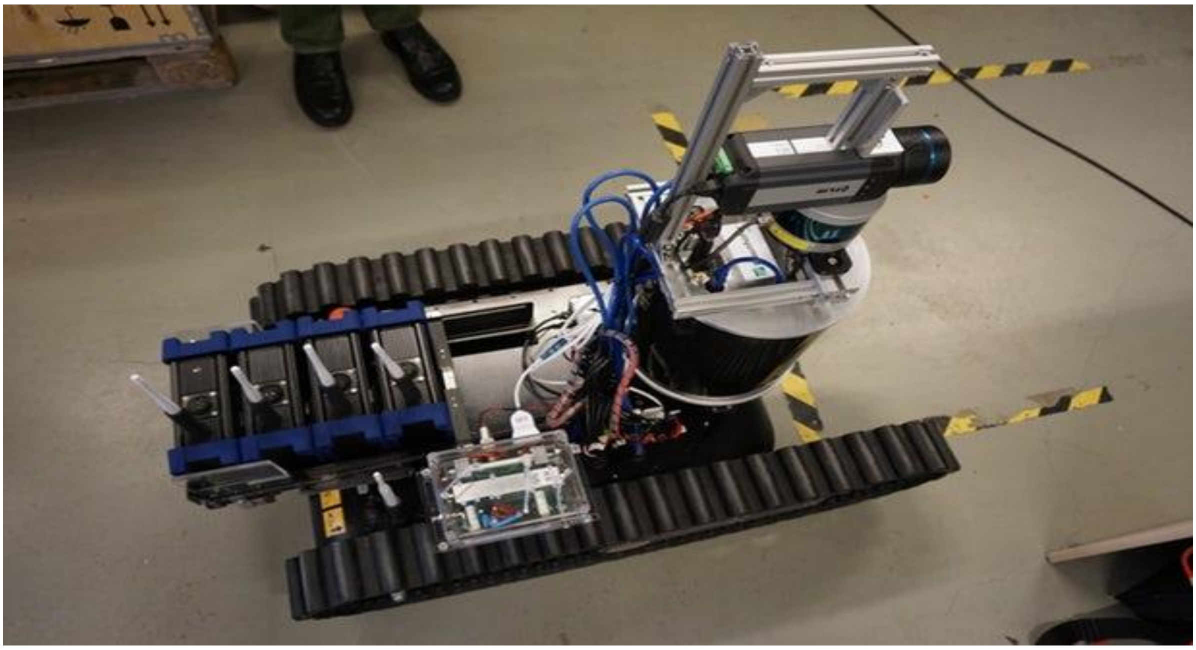

Figure 2. SmokeBot

SmokeBot:

- Designed to operate in low visibility conditions
- Primarily used to assist in fire-fighting scenarios where low visibility and toxic gas are major concerns
- In addition to traditional sensors such as LIDAR and cameras, which are affected by smoke or dust, this robot includes a 3D radar camera, a stereo thermal camera, and high-bandwidth gas sensors
- Strengths:
    - Equipped with many different sensors that work together to map the surrounding area despite low visibility
    - When it loses connection to its human operator, it will automatically navigate back to a previous position that had a better connection
    - Able to detect gas and warn operators if there is a risk of an explosion
- Weaknesses:
    - The additional sensors are heavy, thus it is slow to assist in most critical and urgent rescue missions
    - It takes about 15-30 minutes to collect data, which can further delay rescue efforts
    - Unable to detect people

Section 3 System Architecture

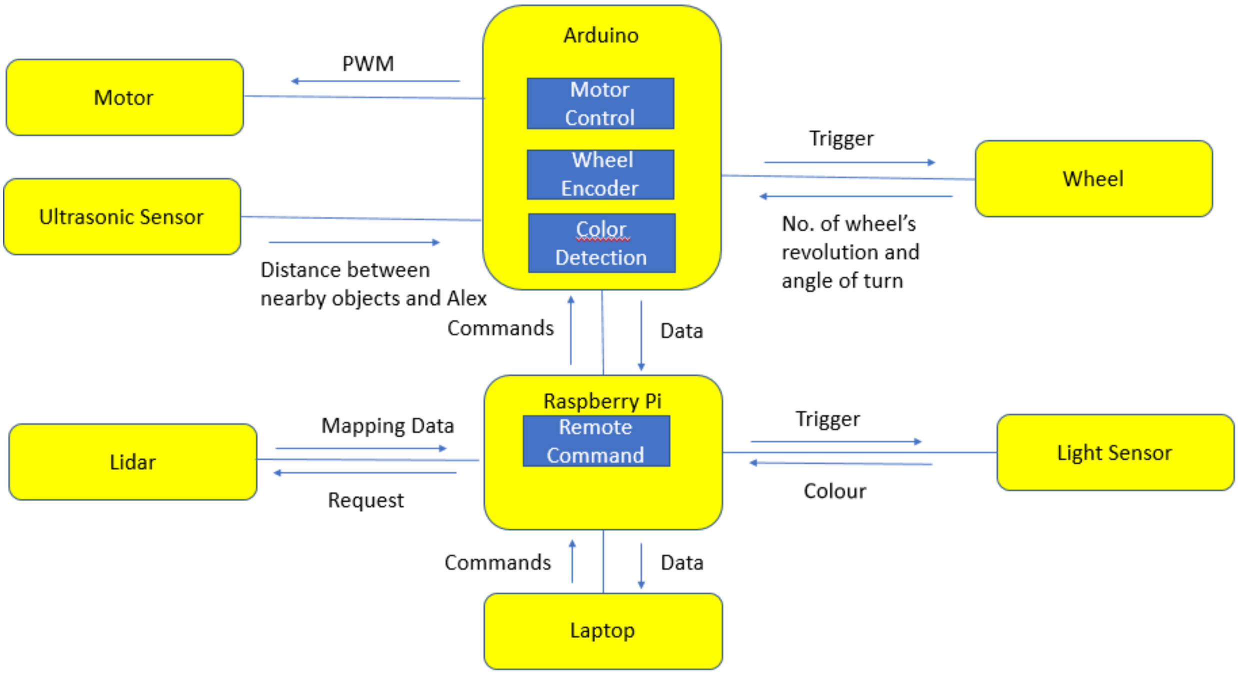

Figure 3. Alex's system architecture diagram

Figure 3 above shows the overall architecture of Alex. The yellow boxes represent hardware while the blue boxes represent software. The arrow between the components represents information / control passes between them.

Arduino:

1. Motor Control---Motor:
    - Controls the speed of the wheel by adjusting the PWM duty cycle.
2. Wheel Encoder---Wheel:
    - When the wheel rotates, the Hall effect on the wheel encoder will trigger an interrupt to track the number of revolutions of the wheel and Alex's angle of turn.
3. Object detection---Ultrasonic Sensor:
    - Ultrasonic Sensor will send data to Arduino on the proximity of any nearby objects with Alex.

Raspberry Pi---Colour Sensor:

1. Colour detection---Ultrasonic Sensor:
    - When Alex detects an object, it will activate the colour sensor and trigger an interrupt to check for the colour of the object

Arduino---Raspberry:

- Raspberry acts as a host that sends commands to Arduino. Arduino returns data (number of revolutions of the wheel and angle of turn).

Raspberry---LIDAR:

- Raspberry sends a request for data. LIDAR receives the request and returns a set of mapping data.

Laptop---Raspberry:

- User sends commands through the Laptop to Raspberry. Raspberry returns data (mapping data, number of revolutions of the wheel and angle of turn)

Section 4 Hardware Design

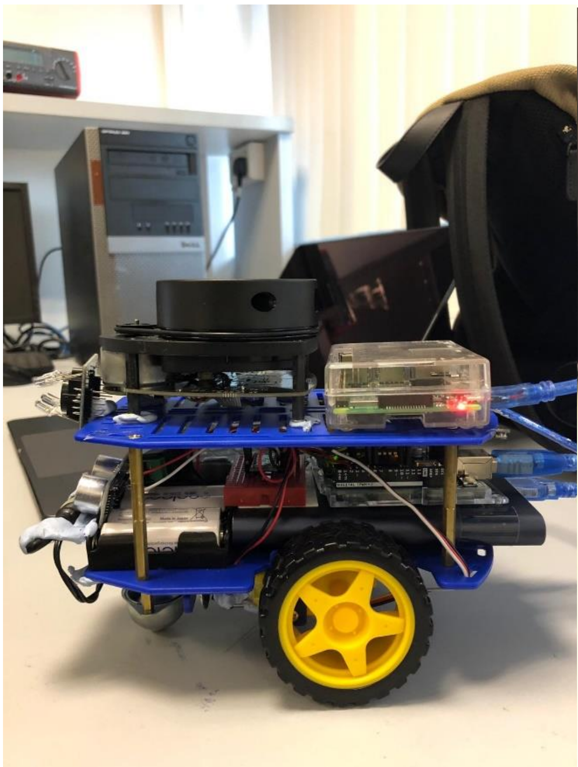

Figure 4. Side view of Alex

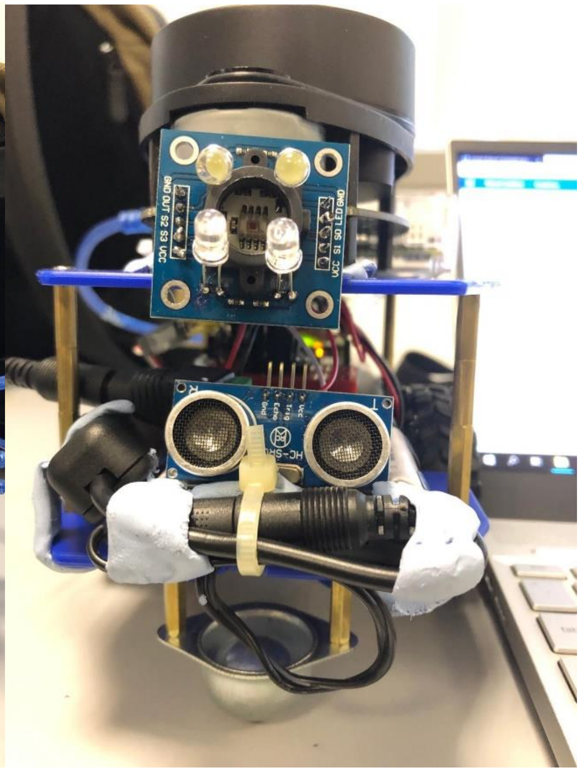

Figure 5. Front view of Alex

- The arduino, breadboard, power bank and battery pack are placed in the chassis so that the wiring is more secure as compared to when placed above to prevent wires from breaking.
- As for the lidar and raspberry pi, they are placed on top, with the lidar in front of the raspberry pi so as to avoid obstruction during the mapping of the surroundings.
- The ultrasonic sensor and colour sensor are held in place using blue tack
- Our team decided to use blue tack as an adhesive to connect the ultrasonic sensor and colour sensor in front of the robot, as well as cable ties to secure any loose wires.

Section 5 Firmware Design

Initialization before maze navigation

a. Raspberry Pi is turned on, and preliminary checks are done. Handshake with Arduino is done.
b. Colour sensor is turned on and calibrated.
c. Motor is tested by the execution of one complete clockwise and counterclockwise turn.
d. LIDAR is tested.
e. Battery power bank is fully charged.
f. Remote into Raspberry Pi using laptop.

Arduino Communication Protocol

We used the Serial port to transmit crucial data between Arduino and Raspberry Pi. These data include direction, distance and angle of turn, which allows us to navigate Alex properly. We connected the Ultrasonic sensor to Arduino to gauge distances that are too short for the LIDAR to completely detect, such as when we are trying to scan for the colour of the bonus objective. We also connected the wheel encoder in order for us to gauge the distance we've travelled along the maze as we move. IT's also necessary to gauge the turning angle of Alex as it traverses the maze.

An example of the high-level steps can be seen below.

High-level steps:

1. *Initialization*
2. *Remote into Raspberry Pi using laptop*
3. *Map out surroundings using RVIZ*
4. *Send navigational commands based on RVIZ data using laptop*
5. *When RVIZ maps out bonus objective, stop Alex*
6. *Activate colour sensor, identify colour of bonus objective*
7. *Repeat steps 3 to 6 until entire maze is mapped and all bonus objectives fulfilled*

Step 1: Initialization (before maze navigation)

a. *Raspberry Pi is turned on, and preliminary checks are done. Handshake with Arduino is done*
b. *Colour sensor is turned on and calibrated*
c. *Motor is tested by the execution of one complete clockwise and counterclockwise turn*
d. *LIDAR is tested*
e. *Battery power bank is fully charged*
f. *Alex is good to go*

Step 2: Send navigational commands using laptop

a. *Movement commands are relayed to the Raspberry Pi through remote computing, which then forwards them to the Arduino through the serial port. The Arduino then translates the commands and controls the motors accordingly*
b. *Stop when an irregular object that may be the bonus objective is mapped out. To be sure, make a full revolution around the objective such that the LIDAR can map out the actual shape of the object.*

Step 3: Activate ultrasonic sensor in our bonus colour objective

a. *Activate front mounted ultrasonic sensor that is connected to the Raspberry Pi. Since we are using the TCS3200 colour sensor which has a built-in LED, the colour sensor will work at optimum performance even in low light conditions.*
b. *Scan the colour for the bonus objective 10 times for each of the colored photodiodes red, blue and green, averaging out the readings. Once the data is obtained, map each of the readings to a range from 0 to 255.*
c. *After mapping, compare the mapped data values of red blue and green, and deduce what is the color of the bonus objective based on the highest RGB value.*
d. *Once the color of the bonus objective has been identified, turn off the color sensor to save power.*

Section 6 Software Design

Raspberry Pi Teleoperation

1. Movement commands are relayed to the Raspberry Pi through remote computing. Using local area network, SSH into the Raspberry Pi using command prompt from a laptop. If Raspberry Pi cannot be on the same local area network as the laptop, port forwarding will be used. Set up RVIZ program on laptop.
2. After relaying the movement commands to the Raspberry Pi, the movement commands will then be forwarded to the Arduino as interrupts.
3. The Arduino then translates the commands and controls the motors accordingly. (See Arduino Communication Protocol above)
4. Stops when an irregular object that may be the bonus objective is mapped out. To be sure, make a full revolution around the objective object such that the LIDAR can map out the actual shape of the object.
5. Our team have decided to use RVIZ program instead of the initial SLAM program because SLAM is power-intensive and using it would require constant re-charging of the battery pack and power bank. Thus, our team is using RVIZ as a temporary substitute which is less power-intensive. The RVIZ is working adequately for the purposes of our project.

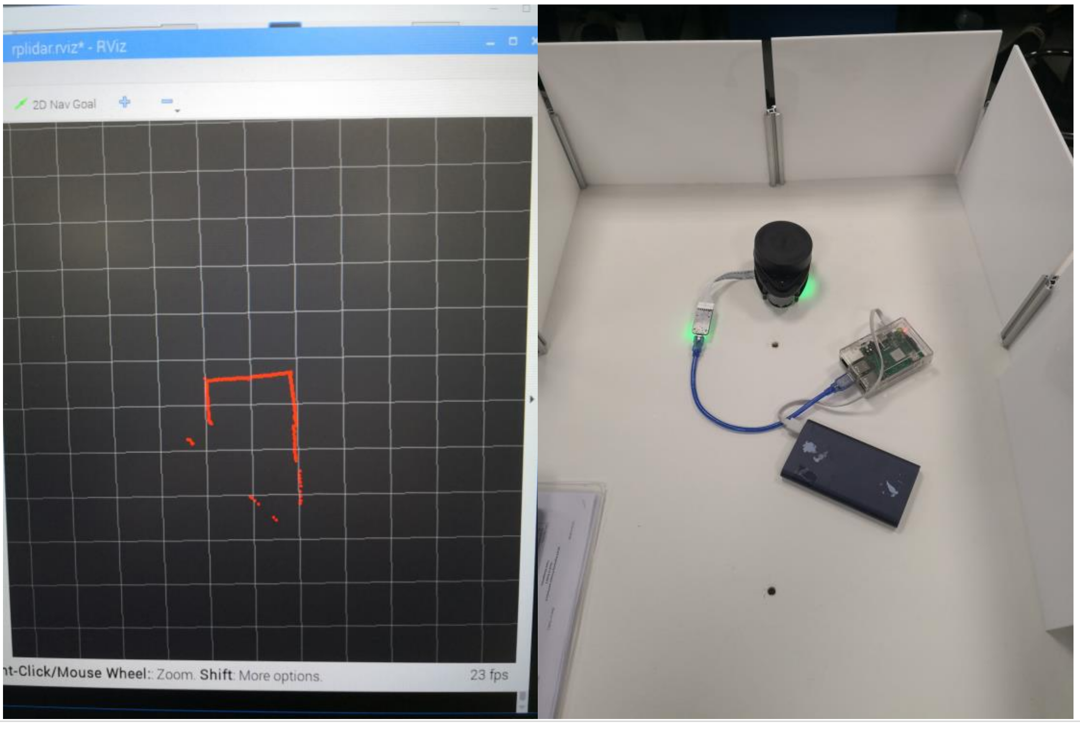

Figure 6. Map returned by LIDAR Figure 7. Actual position of LIDAR

Raspberry Pi Colour detection

1. Activate front mounted colour sensor that is connected to the Arduino by sending a signal from the laptop to Raspberry Pi, which forwards the command to the Arduino as an interrupt.
2. Since we are using the TCS3200 colour sensor which has a built-in LED, the colour sensor will work at optimum performance even in low light conditions.
3. Scan the colour for the bonus objective 10 times for each of the colored photodiodes red, blue and green, averaging out the readings. Once the data is obtained, map each of the readings to a range from 0 to 255.
4. After mapping, compare the mapped data values of red blue and green, and deduce what is the color of the bonus objective based on the highest RGB value.
5. Once the color of the bonus objective has been identified, turn off the color sensor to save power.

Raspberry Pi Ultrasonic sensor

1. When a special objective is identified, we get very close to the bonus objective, as such the lidar can be ineffective
2. Thus, we turn on the ultrasonic sensor, and by tracking the time taken for a fired ultrasonic sound pulse to be reflected back to the sensor, and dividing it by the speed of sound, we will be able to obtain the distance between the robot and the objective
3. Once the distance to the bonus objective has been identified, we will start the approach
4. When we are around 7 cm away from the bonus objective, we turn off the ultrasonic sensor, and switch to the color sensor.
5. Once color identification has been completed, turn on the ultrasonic sensor again, and navigate Alex back towards its original position.
6. Once completed, turn off the ultrasonic sensor, and resume normal operation.

Section 7 Lessons Learnt – Conclusion (All)

Mistake 1:
Our mistake was to rush the design. Lacking in deliberation, it led to many unnecessary rearrangements. For example, the orientation of the robot was initially wrong.

Lesson 1:
Our team realized that proper planning is important, because prior to the final design, we've encountered multiple hassles in rearranging the wires to ensure a more convenient and secured arrangement.

Mistake 2:

Lesson 2:

References

IEEE STANDARD

https://pdfs.semanticscholar.org/d0fb/9b3e69b45b801e887a3f21c7fc112e10c86d.pdf
(RAPOSA)

https://www.smokebot.eu/project.html
(SMOKEBOT)