

**CG1112 Engineering Principle and Practice**

Semester 2 2017/2018

**“Alex to the Rescue”**

**Final Report**

**Team: 02-04-01**

|  |  |  |
| --- | --- | --- |
| Name | Student # | Main Role |
| Claire Chan | A0187908L | Communications between Raspberry Pi and Arduino |
| Evelyn Chen | A0184698E | Arduino “Bare Metal” and Circuit Design for sensors |
| Hashir Zahir | A0132231B | Controls and Colour Detection |
| Quek Aik Peng | A0167696H | Localization and Mapping via Lidar |

**Section 1 Introduction**

72 hours is the “golden period” to search and rescue survivors in any disasters. Alex, our robotic vehicle, is built with the ability of traversing and mapping the environment, as well as identifying the colours of objects of interests.

A network based communications interface, namely the Robot Operating System (ROS), is utilized to communicate between our user computer and the Raspberry Pi (“main brain” of Alex), which the former sends data of interest to our user computer to run the Self-Localization and Mapping algorithms to construct and display the localized map seen by Alex.

On the other side, a communication module is created over USART to interface the Raspberry Pi with the Arduino to control the motors as well as getting data from the infrared (IR) sensors via interrupt service routines and registers. A circuit is built to trigger external interrupts on the Arduino when certain threshold values from the IR sensors are triggered.

The colour detection module makes use of a Pi Camera and image processing to determine whether an object is green or red. The Picam was turned off by default and for most of Alex’s operation in order to save power.

**Section 2 Review of State of the Art**

The main draw of Urban Search and Rescue (USAR) robots is its ability to operate under conditions that are difficult for human rescue teams, accelerating the rescue process during disaster relief efforts where time is of the essence. However, flexibility and robustness are considerations in designing such USAR robots to perform in various disaster scenarios. This section will highlight two such projects that aim to overcome the difficulties in the designs of such robots.

Kinect-powered Search Robot

The robot, designed by students from the University of Warwick, uses Kinect sensors for mapping and navigation [1], instead of the more commonly used LIDAR technology at a lower cost. Kinest sensors were a line of 3D motion sensing devices initially developed for gesture-based games, but is used by the teleoperated robot to map terrain in 3D and transmit data to rescuers remotely, allowing them to identify areas in which victims might be trapped. The robot also has an arm that can be used to manipulate small objects.

However, this robot is only designed to search, and is yet equipped with rescue functionalities. Furthermore, this robot is still a prototype and has several insufficient systems attached [2]. While there were not any further developments on this particular project after 2013, the use of the Kinect to map disaster zones has been picked up in other USAR robot projects, such as cockroach-like robots piloted by Kinect [3].

Snakebot for urban search-and-rescue

This serpentine search robot, or “snakebot”, consists of several modular joints that produce a range of motion, most notably imitating the movement of a snake, allowing it to traverse varying environments and crawl into small confined spaces that are otherwise inaccessible to humans [4].

Lights and cameras are attached to the head, while other sensors are packed in its body, allowing the robot is to broadcast captured footage to rescuers. The modular design means that it consists of connected independent modules working together for the snakebot’s continuation of its operation even with a few damaged segments [5].

However, the usage of the robot requires at least four operators, and is not easy to operate [6]. Furthermore, rescuers on the ground rely more on sound and smell (compared to sight) to search for victims, but the snake robot currently lacks sensors such as microphones and speakers to collect such information for the rescuers [7].

**Section 3 System Architecture**

As shown in the figure below, communications happen between many devices. The Raspberry Pi and the user computer communicate through the ROS, while the Arduino and the Raspberry Pi communicate through USART. Data is transferred through these various ways, while commands are sent in order to retrieve these data.

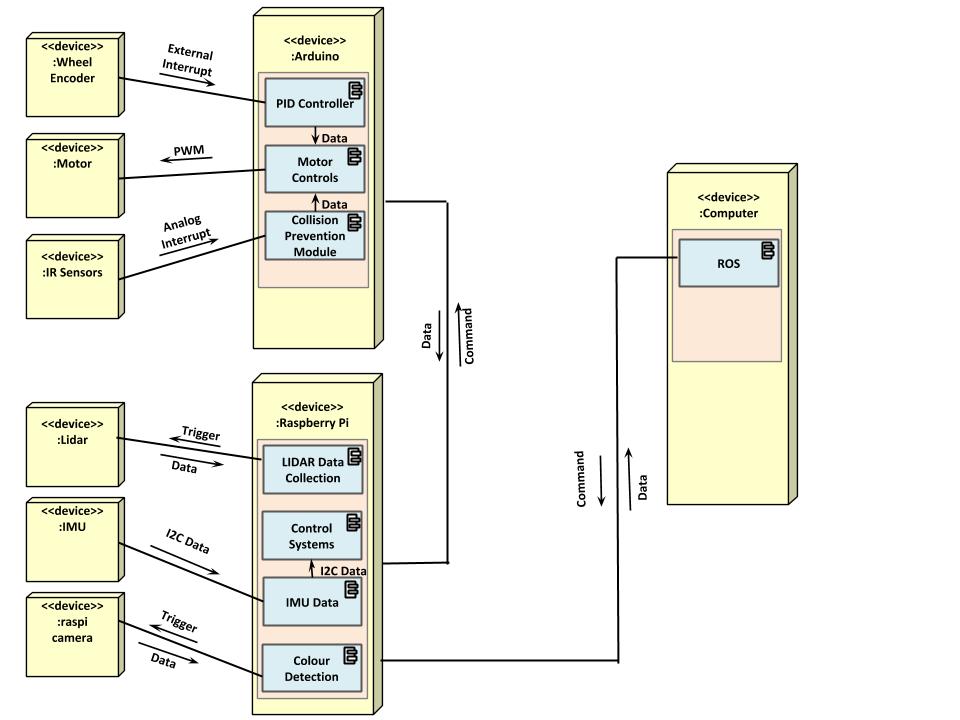
****

Figure 1. UML deployment diagram of Alex

**Section 4 Hardware Design**

The below figure provides a front and back view of the final form of the system, indicating the placement of important hardware components. The LIDAR is used to map out the environment, while the Pi Cam is used to detect the colour of the mysterious object. On the other hand, the IR sensors are used to detect the distance between Alex and surrounding obstacles or walls. The Power Bank supplies power to the Raspberry Pi, while the Battery Pack supplies power to the Arduino.

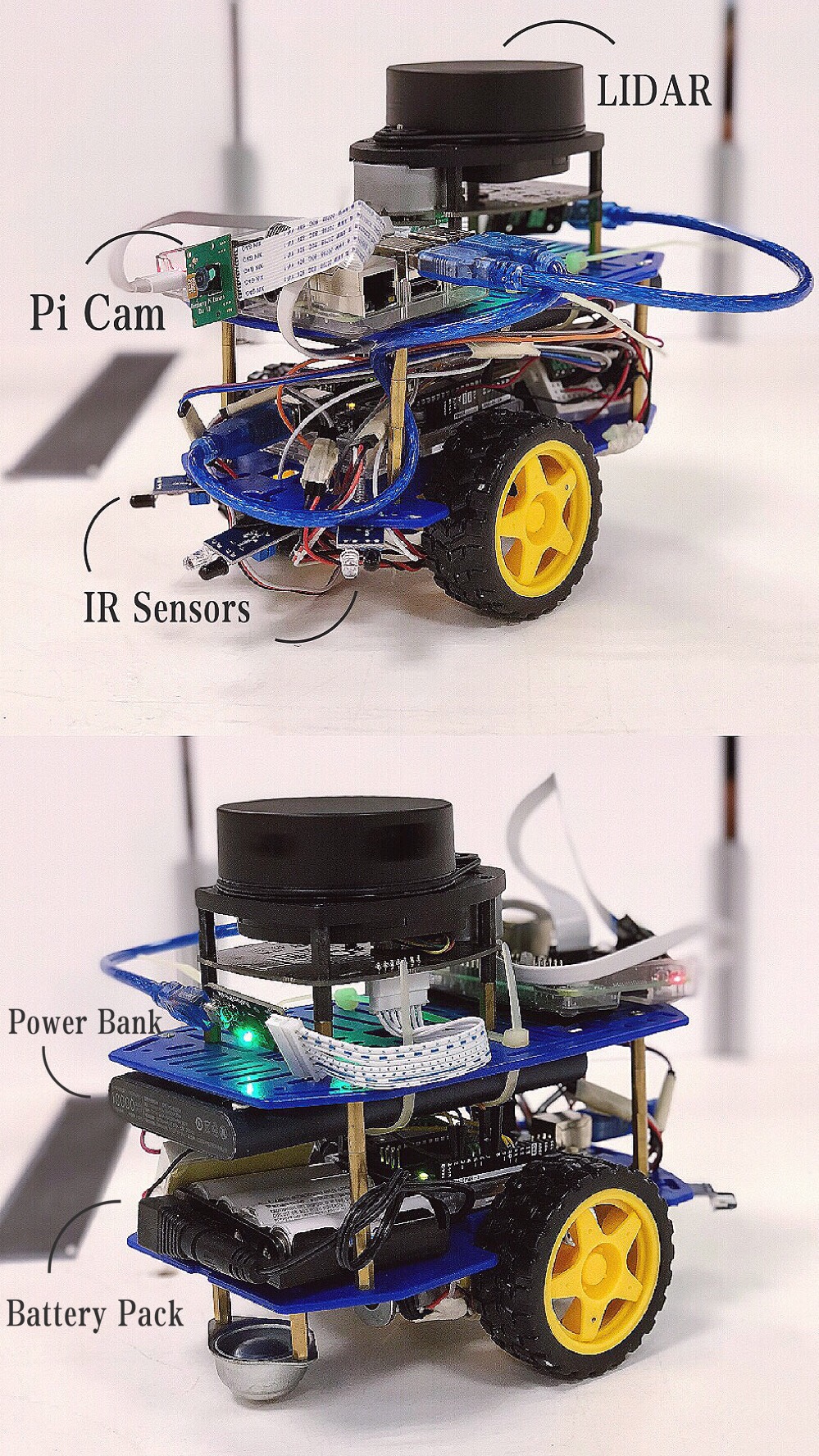


Figure 2. Final Form of Alex

**Section 5 Firmware Design**

**Movement**

The Arduino first activates the motors and rotates Alex in the desired orientation of movement. While this happens, the Pi will take readings from the IMU module, and will instruct the Arduino to stop the motors once the desired orientation of Alex is achieved. Following which, the Arduino activates both motors to move Alex forward. Using data from the wheel encoders, the Arduino counts the pulses received to determine the distance that has been travelled. The Arduino will concurrently use feedback from the PID module to ensure that Alex moves as straight as possible. Once the desired distance is achieved, the Arduino will stop the motors.

**Emergency Stop**

In the event that the user makes a mistake, Alex will continue moving forward until it receives an interrupt from the collision prevention module to stop all motors. The collision prevention module, comprising mainly of IR sensors, is each made of a simple circuit with a comparator to output digital low - when an object is far away - and digital high - when an object is near. This is done so that the Arduino will receive an interrupt when the front section of Alex is within a threshold distance, for example, 5 cm from an obstacle. When such a situation occurs, the analog voltage output from the IR sensor would cross a threshold voltage. Hence, The Arduino will be able to pick up this change in logic level via interrupts and stop the motors, preventing Alex from crashing while moving. A message will also be sent from the Arduino to Raspberry Pi to inform the Pi that Alex has stopped moving. This will then be conveyed via ROS to the human controller on the user computer. The user will subsequently need to orientate the vehicle away from the obstacle and so that Alex has a free path to maneuver along.

**Map out surroundings**

While Alex moves, the Pi will continually request the LIDAR for a reading of the environment at specific intervals. Given that the LIDAR operates on a rotating laser light, the motor on the LIDAR requires some time to rotate the laser light by 360 degrees. Additionally, more time is needed for this data to be captured and transmitted by the LIDAR over USART to the Raspberry Pi. This whole process of collecting 1 sample of 360 degrees of data takes a while, and therefore it is recommended to advance Alex slowly. A low movement speed allows the collection of LIDAR data that can represent a complete map of the surroundings without unfilled gaps and unmapped points. The collected raw LIDAR data will then be transferred over ROS (/lidar) to our computer where SLAM will be performed to piece together the localized map as seen by Alex.

**Section 6 Software Design**

**Initialisation**

We first establish a connection from our user computer to the ROS master on the Raspberry Pi. The connection is established if we receive a success status reply from the Pi when we send a message asking about its status. Afterwards, we send a custom message to a specific topic in ROS to initialize all subsystems. The IMU Magnetometer would be zeroed with its current heading being true north, followed by controls module initialized, all motors are stopped, IR sensors initialized. Last but not least, data will start to steam to the collision prevention module. All the data will be streamed to the Pi, and then back to the user computer, to let us know that all systems are ready.

**Transmit data to use**

After every command, the Arduino will send a message to the user computer that the command has been completed successfully (or not in the case of emergency stop). The Pi would then transmit this data, along with the current IMU data, back to the user so that the user can make an informed decision for the next command.

As Alex traverses the terrain, raw LIDAR data will be continuously streamed to our user computer via ROS in short intervals, constantly updating the local map. Once Alex has fully explored all 4 rooms in the challenge, we would have met the primary requirement of mapping the entire surroundings. Throughout Alex’s journey, we will also keep a lookout for regularly shaped objects to scan for the secondary requirement.

**Colour Detection**

The colour detection module on the Raspberry Pi will first switch on the Picam to analyse the object in front of it, then ran on the raspberry pi by listening to the raw image topic published by the camera, for example, /camera/image\_raw. This will then be followed by several simple OpenCV colour thresholding algorithms to determine whether the object is green or red. For example, Gaussian Blur will be used to smoothen the image and reduce noise. RGB Colour thresholding can be done to generate a binary image of green or red objects after which contour detection will be performed to calculate the area of the object to ensure that it is not some random noise picked up, for example, reflections of red or green object, and that the object detected is large enough. The predicted colour will then be published by the colour detection module, for example, /object\_colour, and the user computer can subscribe to that topic to receive information on the colour of the object or lack thereof detected by Alex.

**Section 7 Lessons Learnt - Conclusion**

Like many other groups, our left and right motors were not of the same power to begin with. One would be moving freely while the other is stuck and makes clicky sounds when manually turned. Besides, their wheel encoders occasionally neglect to send any values back when it was clearly supposed to return us numbers. The major difficulty during this entire project was that our motors and wheel encoders stopped working numerous times during this project, leaving us in endless despair, especially when it was coming to Week 12. Each and every time this problem occurs, it takes us several hours and attempts to fix it. Through many struggles, our biggest lesson learnt was to connect wires properly. Do not give them any chance to come loose or out-of-place. Another lesson learnt was also regarding wires, one ought to replace wires if they have the potential of causing problems.

On the other hand, as a team, we learnt that it is difficult to follow our proposed schedules of completion when we all have other assignments and exams to worry about. What we could do in future projects could be setting more realistic goals and having a detailed list of achievements that must meet by certain deadlines. Furthermore, it is essential to realize each and every members’ strengths and weaknesses to optimise our production rate.

**References**

[1] D. Lee, “Microsoft Kinect-powered robot to aid earthquake rescue”, *BBC News*, Feb. 2011. [Online]. Available: https://www.bbc.com/news/technology-12559231.

[2] K. Buckstone et al., “Warwick Mobile Robotics: Urban Search and Rescue Robot”, University of Warwick, May 2013.

[3] E. Moore, “Future of search and rescue: Cockroaches piloted by Kinect”, *CNET*, June 2013. [Online]. Available: https://www.cnet.com/news/future-of-search-and-rescue-cockroaches-piloted-by-kinect/.

[4] G. Nichols, “Snakebot named ground rescue robot of the year”, *ZDNet*, April 2018. [Online]. Available: https://www.zdnet.com/article/snakebot-named-ground-rescue-robot-of-the-year/.

[5] M. Crawford, “Snake Robots Crawl to the Rescue, Part 1”, *ASME*, May 2018. [Online]. Available: https://www.asme.org/engineering-topics/articles/robotics/snake-robots-crawl-rescue-part-1.

[6] M. Hutson, “Searching for survivors of the Mexico earthquake - with snake robots”, *Science*, Oct. 2017. [Online]. Available: https://www.sciencemag.org/news/2017/10/searching-survivors-mexico-earthquake-snake-robots?r3f\_986.

[7] E. Ackerman, “What CMU’s Snake Robot Team Learned While Searching for Mexican Earthquake Survivors”, *IEEE Spectrum*, Oct. 2017. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/industrial-robots/cmu-snake-robot-mexico-earthquake.