CG2111A Engineering Principle and Practice II
Semester 2 2022/2023

"Alex to the Rescue"
Design Report
Team: B01-6A

Name	Student #	Sub-Team	Role
Jonathan Ong	A0283070L	Software	Colour Detection/ Hardware Design
Quan Teng Wai	A0281416H	Software	LIDAR Implementation
Piraveen Alexander	A0273022U	Software	Robot Movement
Chan Zun Mun Terence	A0282142M	Hardware	Circuit Design

Section 1 System Functionalities

Our Alex Robot is designed to emulate the functionalities of a search and rescue robotic vehicle. It would be wirelessly controlled by operators to move in a simulated environment.

Alex will include a Raspberry Pi and an Arduino Mega. The Raspberry Pi has a Master Control Program to communicate with Arduino via UART. The Arduino is then able to control the motors and return information from the ultrasonic sensors to the Raspberry Pi, for Alex to gather data and navigate through the Course. At the same time, the Raspberry Pi receives data from the LIDAR to map out the environment.

The operators would use a Laptop. It communicates with the Raspberry Pi to remotely control the robot, using SSH for controlling movement & sensors, and ROS Networking for retrieving SLAM data.

<u>Simulated Environment</u>
Our Alex would be used in a room with various objects, such as tables, chairs and boxes. We would pilot Alex to find at least 2 victims in a 6-minute window. These victims are green (healthy but trapped), or red (injured and trapped).

<u>Main Functionalities:</u>

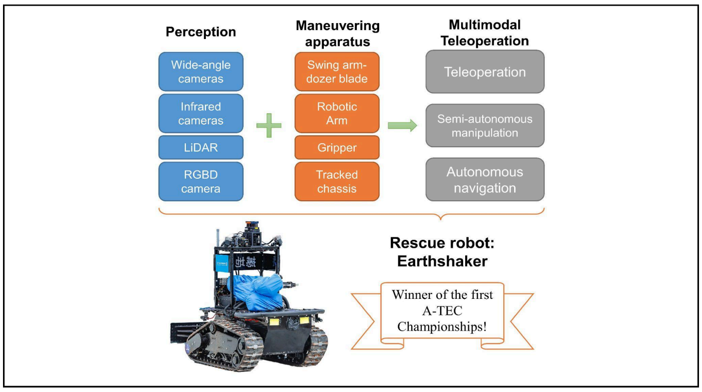
1. Move orthogonally (Forwards, Backwards, Left Turn, Right Turn)
2. Identify objects within the test area using the Colour Sensor
3. Map out the test area with the help of the LIDAR and SLAM
4. Enough battery life to minimally last the entire test duration (~15min)
5. Be remote-controlled by a teleoperator

<u>Additional Functionalities:</u>

1. Detect the colour of any nearby object.
2. Alex is able to be “cool”
    a. It would have Ultrasonic Sensors to detect the distance of the front/ back/ side of the robot to the obstacles
    b. It would have a face (our teammate’s)

Section 2 Review of State of the Art

There are many different designs in the modern search and rescue robotic arsenal. Here are 2 that interested us.



1. Earthshaker, winner of the first Advanced Technology and Engineering Challenge (A-TEC) championships.

Teleoperation was achieved with 3 different communication methods between the Operator PC and the On-board computer (NUC).
Using a 1.8G MIMO mesh radio, 2xAT9S, STM32 Control board and a 4G/5G router.
Software enables automated switching between communication modes depending on connection quality/ range.

Strengths	Weaknesses
• Chassis and components minimally IP64 water resistant, able to handle wet environments • Dozer Blade allows it to push obstacles under 75kg away • Multiple communication methods, from 2.4Ghz direct communication to 1.8 GHz MIMO-mesh radios to suit a different ranges and conditions	• Excessive size limited flexibility of movement • Lacking in gripping power for modified robot arm, unable to manipulate heavier loads. • Battery life of 3h • Cumbersome body slower at navigating stairs

2. Vapor 55 Drone, used in spotting sharks, dropping life rafts and medical equipment in Australia, originally designed to deliver blood and supplies to wounded soldiers.

Hardware:
Trillium HD-25 Optical sensor (Camera)
Lidar/ PPK Mapping Options

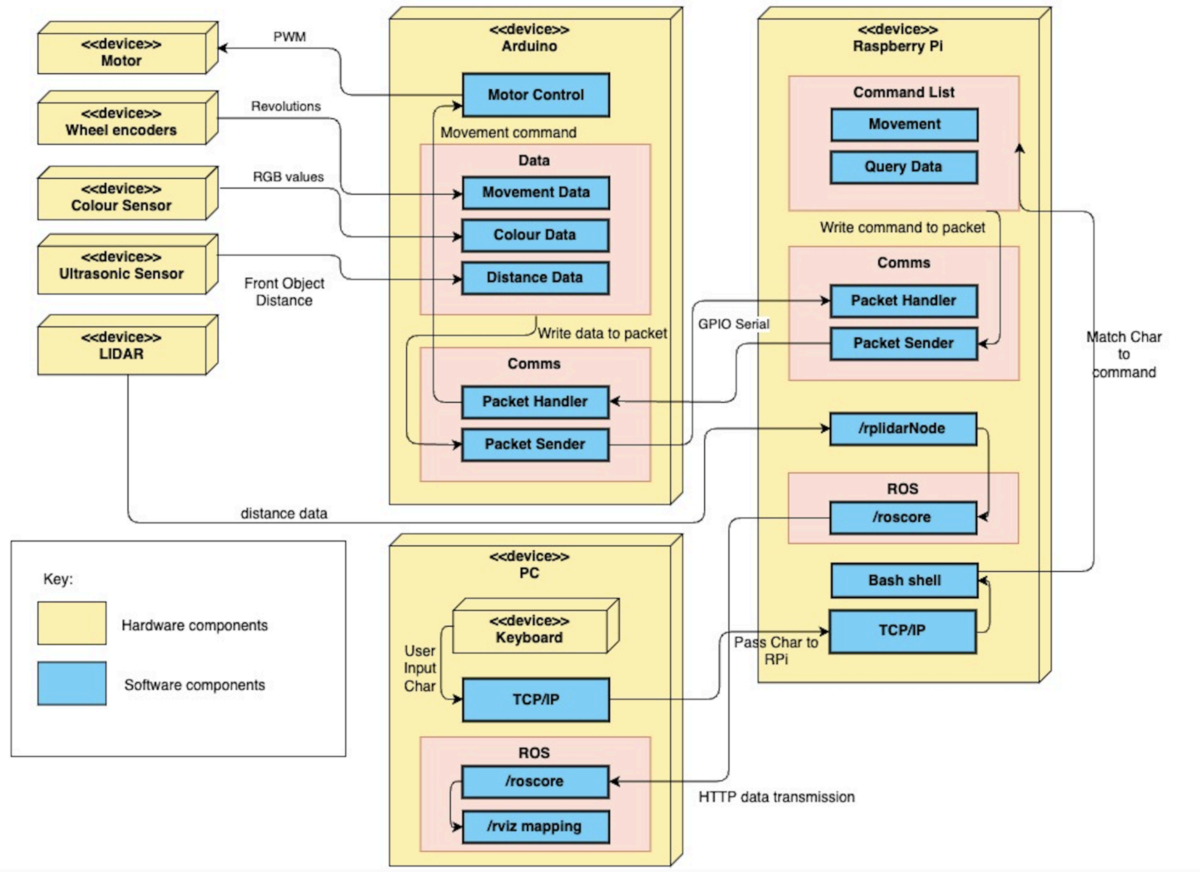
Software:
Advanced Flight Control System (FCS) [inclusive of autopilot]
Automatic Autorotation, smart auto-home functions

Strengths	Weaknesses
• Modular radio options (Common Radio Interface connector) • Flexible Payload - rail design enables many different kinds of payloads to be equipped • Aerial - Unrestricted by Terrain	• Battery life of 1h • Environmental Operational Limits from -17C to 49C, unable to be used in extreme temperatures • Comparatively low Usable payload of 4.5kg compared to ground options

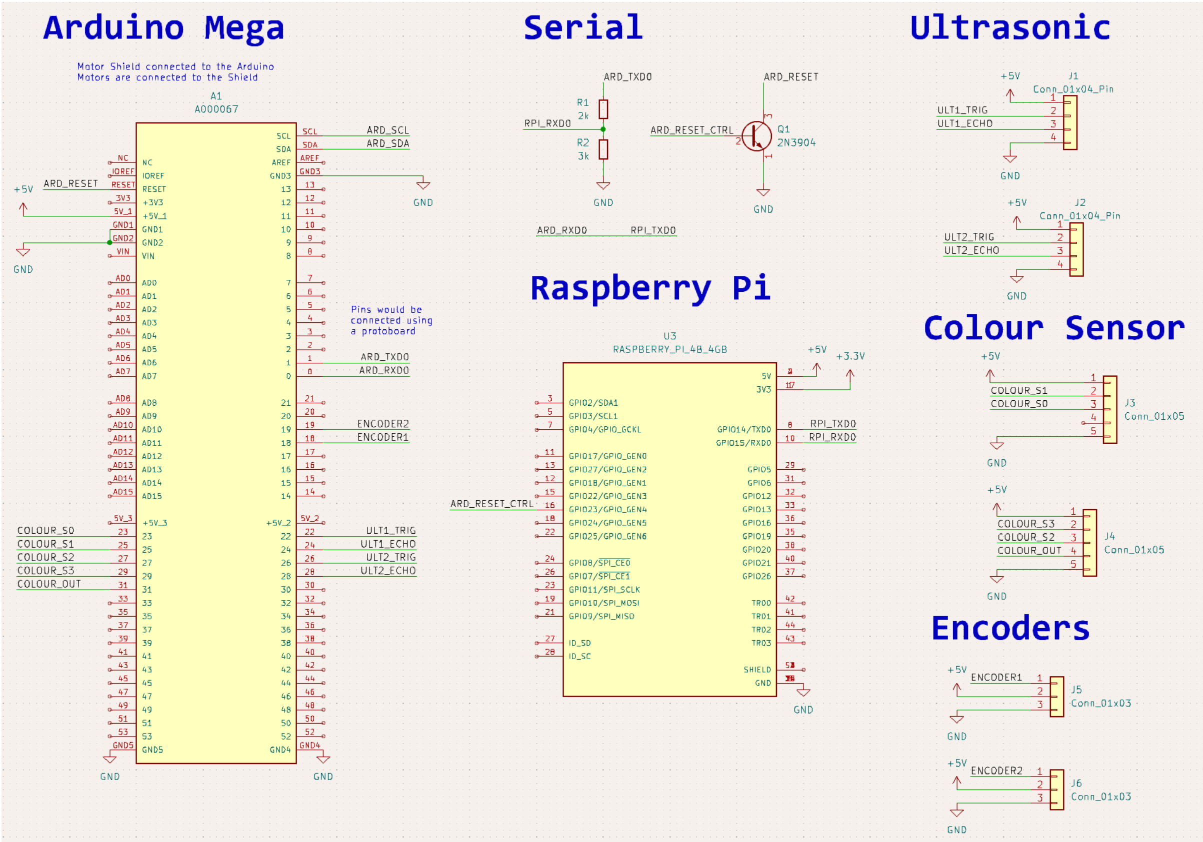
(Rubble/ Ocean)	

Section 3 System Architecture

High Level Components



Schematic (Subject to Changes)



Electronics will be connected through wires and a protoboard for ease of assembly, and ease of access to the pins.

1. LIDAR would be connected to the Raspberry Pi via USB
2. Motors would be connected to the Motor Shield, and the Motor Shield would be connected to the Arduino. The Protoboard goes between the Shield and the Arduino
3. The RPi would be powered by a USB Power Bank, while the Motors and Motor Shield will be powered by 4 1.5V AA batteries

Section 4 Component Design

High-Level Algorithm Steps:
1. Initialise communication between Arduino, Pi and PC
2. Generate map with Hector SLAM
3. Receive tele-operator commands
4. Carry out commands
5. Repeat steps 3 and 4 until navigation is over

Further Breakdown

Step 1: Initialise communication between Arduino, Pi and PC

a. Pi - Arduino Mega Communication
    i. Pi opens a Serial Connection to the Arduino Mega
    ii. Pi creates, serializes and sends a Hello Packet to the Arduino
    iii. Arduino initialises the Serial port and responds to the Hello Packet by sending back an OK packet
b. Pi - Laptop Communications
    i. Pi opens a TCP/IP server and binds to a port
    ii. The laptop connects to the server over the same WiFi Network
    iii. With the help of OpenSSL, the Laptop authenticates with the Pi and communicates securely.
c. Others
    i. ROS is Started on the Pi, which the Laptop connects to through ROS networking
    ii. The Laptop can SSH/ VNC into the Pi

Step 2: Generate map with Hector SLAM

a. Pi gets distance data from LIDAR and odometry data from Arduino
b. Pi sends occupancy data to PC through ROS
c. PC generates map with Hector SLAM with occupancy data

Step 3: Receive tele-operator commands

a. After reviewing map, tele-operator sends commands:

Movement Commands				
W	**A**	**S**	**D**	**X**
Forward (increase linear speed in positive direction)	Rotate Left (increase angular speed in negative direction)	Stop	Rotate Right (increase angular speed in positive direction)	Reverse (increase linear speed in negative direction)

Additional Commands	
C	**V**
Sense colour	Get ultrasonic sensor distance

b. PC sends commands to Pi through TCP/IP through the packet sender
c. Pi handles packets and sends commands to Arduino

Step 4: Carry out commands

a. Depending on the command, Arduino does the following:

Command Type	**Movement**	**Colour**	**Ultrasonic Sensor**
Execution	sends PWM signals to motors	triggers colour sensor	triggers ultrasonic sensor
Return	odometry data calculated from wheel encoders	colour determined from RGB values	returns ultrasonic distance

b. Pi receives the data from the Arduino and LIDAR and handles it as follows:

Arduino		LIDAR
Colour	Ultrasonic Sensor	Map data
Returns data to PC through **TCP/IP**		Publishes to **ROS** topic

c. PC receives data and does the following:

TCP/IP data	ROS data
displays to tele-operator	updates Hector SLAM map and displays to tele-operator

Step 5: Repeat steps 3 and 4 until navigation is over

a. Tele-operator identifies when navigation is over and sends command to end program

<u>Serial Communications</u>

Raspberry Pi

1. Pi opens a Serial Connection to the Arduino Mega
    a. Baud Rate 9600 bps, 8N1 Frame format
2. Pi creates a receiver thread to receive packets from Arduino
    a. The packet is deserialized into a TPacket packet, and TResult result
    b. It checks TResult for the status of the packet
        i. If the Packet received is OK, it handles it (such as in Further Breakdown Step 4b)
        ii. Else, it displays the error, be it an unknown error, bad magic number, or bad checksum, and sends it to the laptop
3. Pi Sends command packets to the Arduino depending on the teleoperator commands
    a. It creates a TPacket commandPacket
    b. Configures the packet command and parameters

c. Serialises it into a TComms packet
d. Sends it over to the Arduino through UART

Arduino Mega

4. Arduino Mega sets up Serial port
    a. Baud Rate 9600 bps, 8N1 Frame format
5. Arduino polls for serial data
    a. It reads the Serial data & stores it in a buffer
    b. The data is then deserialized into a TPacket packet and TResult result
    c. It checks TResult result for the status of the packet
        i. If the Packet received is OK,
            1. it handles it (such as in Further Breakdown Step 4a),
            2. It sends an OK packet to the Pi, or a packet with the required information depending on the command.
        ii. Else, it sends an error back to the Pi, be it an unknown error, bad magic number, or bad checksum
        iii. The creation of packets is similar to how the Raspberry Pi does it

<u>Distance Sensing Algorithm</u>

1. Set the trigger pin to high for 10 microseconds to send the pulse
2. Reads the Echo Pin and gets the length of the pulse on the echo pin
3. Convert the period to the distance between the ultrasonic sensor to the wall
    a. Distance = (*Speed of Sound* × *Time take*n) / 2
4. Sends a packet with the distance information to the RPi

<u>Color Sensing Algorithm</u>

1. Set Trigger pin to HIGH to trigger Colour Sensing
2. Colour Sensor Module to cycle between RGB lighting
    a. Individually Toggle Each of the colour LED Pins to HIGH, delay, then LOW.
    b. Read the respective frequencies from the colour sensor during each colour phase
3. With reference to the ROYGBIV table that has the respective range of frequencies for each colour, determine the colour of the object in front of the Color Sensor.
4. Relay information to the RPi, which is then sent to the laptop.

Section 5 Project Plan

Week	Hardware	Software
8 & 9	Assembling Alex & Electronics	1. Communications between Arduino & Raspberry Pi 2. Configure Motor Encoders 3. Set up Secure Communications (TLS)
10	1. Finalise Electronics Schematic & Wiring the Protoboard 2. Debugging the Protoboard & Electronics	Setting up the Raspberry Pi 1. SSH, VNC remote protocols for Debugging 2. Code Repository & Layout
11	1. Mounting of Components 2. Designing and 3D printing required structures	1. Setting up Hector SLAM 2. Setting up Colour Sensor & Colour Detection Algorithm 3. Test Basic Movement
12	1. Adjusting Mounts	1. Setting up Ultrasonic Sensor 2. Calibration of Motors, Ultrasonic Sensors and Colour Sensors
13	1. Final Calibration & Further improvements based on trial run 2. Project Runs	
Reading Week	Final Design Report	

References

1. Zhang Y, Li Y X, Zhang H F, et al.
   Earthshaker: A mobile rescue robot for emergencies and disasters through teleoperation and autonomous navigation.
   JUSTC, 2023, 53(1): 4 doi: 10.52396/JUSTC-2022-0066

2. Avinc (n.d.) *Product catalog*. Retrieved March 28, 2024, from https://www.avinc.com/images/uploads/product_docs/ProductCatalog.pdf

3. Avnic (August 30, 2019). *Vapor55 Datasheet*. Retrieved March 28, 2024, from, https://www.avinc.com/images/uploads/product_docs/VAPOR55_datasheet_08302019.pdf

4. Little Ripper (n.d.) *Our Amazingly Awesome & Proven UAV Fleet: Westpac Lifesaver Series.* Retrieved March 28, 2024, from, https://littleripper.com/our-fleet/