5XIB0 - DBL Venus Exploration

Design Report

Team 23

Full Name	Student ID
Adam Lasota	1793411
Ana Sirbu	1829858
Bastiaan Schaap	1499386
Daniel Tyukov	1819283
Jiahui Que	1886339
Ioanna Panagiotopoulou	1785729
Malik Weren	1788701
Martijn Oosterhuis	1860615
Stefano Bracciali	1880330
Steven Zollinger	1889818

Hand-in deadline: May 12, 2023

Eindhoven, July 9, 2023

Contents

1 Problem Statement

Robots equipped with multiple sensors and actuators are expected to explore the planet Venus and navigate through a maze of steep hills and craters to find three hidden rock samples and bring them back to the predefined lab while staying within the boundaries of the map. The robot has to avoid the hills and the craters at all costs, or else the robot will be lost. This exploration mission comes with several challenging tasks. The key challenges can be formulated as follows:

- Navigate around the planet without colliding with any obstacles (hills and craters).
- Detect and collect the rock samples.
- Navigate back to the laboratory with the rock samples, again without colliding with any obstacles.

In order to overcome these main challenges, the robot is equipped with two servo motors that drive the wheels, a servo motor that operates the gripper (up and down, open and close), and an ultrasound sensor as a baseline. Additional sensors and actuators can be added to improve the functionality of the robot. The added components will consist of infrared sensors, which will detect the black lines around the craters, and a main beacon at the center of the laboratory, combined with smaller beacons on top of the robots to get them back to the laboratory.
This way, the hills can be detected by the ultrasound sensor, the crates, the map limits, and the rock samples can be detected by the infrared sensors and the beacons, combined with a compass software module, will make sure that the robots can navigate back to the laboratory.

2 Discussion on system-level design

The robot has to make the right decisions in order to find the rock samples and return them to the lab. In broad lines, the robot has to be able to perform the following steps:

- Leave the laboratory
- Drive around to be able to search for the block and retrieve it
- Use infrared and ultrasound sensors to scan for obstacles, in order to avoid them
- Use infrared sensors to locate the rock sample
- Pick up the rock sample with the gripper in order to retrieve it
- Locate the laboratory when the rock sample is picked up
- Find the entrance of the laboratory
- Drive in and deposit the rock sample

First of all, it is needed to make the robot to drive. So we can start by programming the wheel encoders in order to drive forward for a specific distance and then stop. If the robot detects an obstacle it has to turn always the same angle to left or right, which has to be less than 60 degrees, approximately, so as to not get stuck in a loop, but randomizing will take longer to reach the goal. In order for the robot to be able to avoid the obstacles, after driving this specific distance of approximately 10cm, Arduino tells the infrared and ultrasonic sensors to scan for obstacles. If one of the two sensors detects an obstacle, then Arduino tells the wheel encoders to turn, accordingly to the instructions given to them and respectively, and if no obstacles are detected, then Arduino tells the wheel encoders to drive forward for this specific distance and then stop again.

The goal of the project in its entirety relies on a working path-finding algorithm. The algorithm must be able to avoid the obstacles, which in this case are hills and craters. The hills have white tape and consist of a rigid material that reflects ultrasound and they will have a height of approximately 30cm, according to the given information from the video. This means that the algorithm will need to use the input from the

ultrasound sensor to avoid them. On the contrary, the craters will be black tape on the ground, which makes the IR sensor perfect to prevent them. A possible way of navigating is to save the path that is used to find a sample. With this, the path to the original stationary position of the robot can be taken in the opposite direction to return to the 'lab'. However, this would take a lot of time, so the solution to this would be to save only the original stationary position. Using this method the robot can get back the fastest. If there are craters or hills in the way, it will go around and will look for the fastest path back.

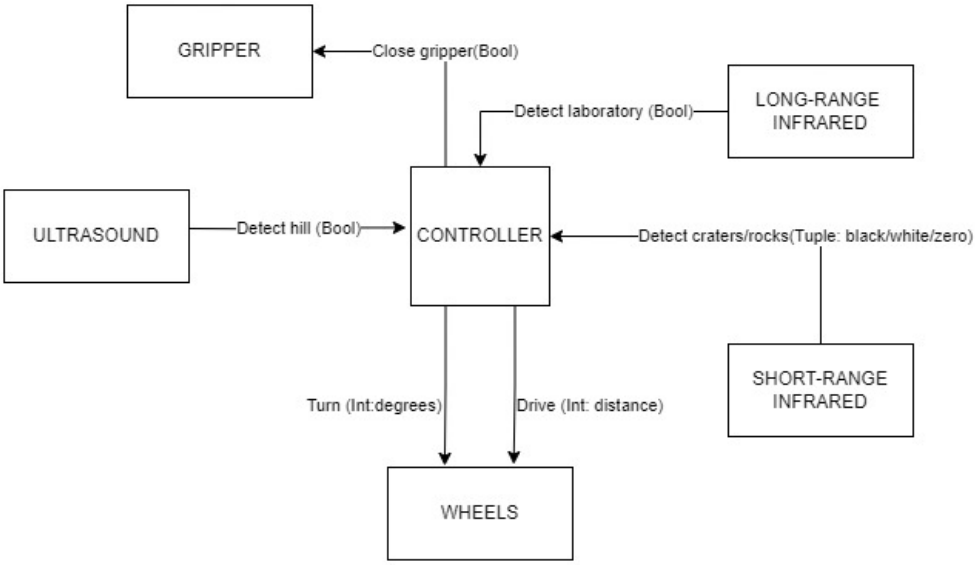


Figure 2.1: System interface between the components

As we can see in Figure 2.1, everything is connected to the controller. It receives information from:

- The long-range infrared, for detecting the laboratory;
- The short-range infrared, for detecting craters and rocks;
- The ultrasound, for detecting hills.

And it gives information to:

- The gripper, to toggle its state between open and closed;
- The wheels, used to either drive or turn.

2.1 Design Alternatives

2.1.1 Infrared Detection

- For the infrared receiver part of the circuit we had to choose between three different components. One of the alternatives is a photo-resistor, but the response time is too slow for our purpose. The other alternative is a photo-diode, which is also a viable option, but the sensitivity of this component is too low. So we ended up using a photo-transistor for our circuit, which has a good response time and the circuit is simple to build.
- If we want to improve a bit more our circuit we can add a frequency to the transmitter and a high-pass filter to the photo-transistor, so as to eliminate noise. Moreover, we can add more IR emitters per sensor to increase the intensity of the infrared light or we can try to add a operational amplifier to amplify the signal.
- If this did not work we were thinking of using a pre-made infrared sensor from Huub (Flux 0.088), but the maximum distance of this sensor is around 0.5 mm, which is not enough for our purpose.

2.1.2 Mounts and Movement

- Should the infrared sensor for detecting the rock sample face straight, down, or at an angle? Facing straight it could detect the rock from far away, but making it more difficult to know when to engage the gripper, but if the infrared sensor is faced downwards it would be the other way around. So for now we settled with positioning the sensor downwards.
- We discussed if we want to let the robot drive a certain distance before scanning the surroundings. If we go by this strategy we would need to choose a distance for the robot to drive before checking. Further testing would be needed to determine the right distance for optimal efficiency, we expect it to be around 10 cm.
- We discussed how many degrees the robot should turn when encountering a hill, crater, or border. Should it turn 30 degrees and check the sensors again or more? If it would only turn 30 degrees it could take longer to clear obstacles. If the robot turns more than 30 it might not follow the mountain properly and may overshoot. For this reason, we chose a turning range of 30 degrees for now, before further testing.
- There will be a mount on top of the robot with infrared receivers. How high this tower needs to be for optimal efficiency to pick up the infrared light from the lab beacon still needs to be determined.

2.1.3 Ultrasound

- We chose to make the ultrasound sensor static rather than dynamic (by not using the servo). This choice implies a limited scanning of the surroundings, but it also avoids the complexity of writing an algorithm for the movement of the sensor and implementing it in the whole system.
- We chose to run the sensor detection output through a filter in order to ensure accuracy. By considering each 4 consecutive outputs as a single output value we can rule out the random errors that may occur. We chose to apply this filter after noticing during tests that, throughout the boolean output values of the sensor, there are a number of outliers that can interfere with the overall movement of the robot.

2.1.4 Laboratory Beacon

- Different frequencies for the PWM signal were considered to be inputted. After doing research, 38kHz will be the best frequency for this application, as Arduino Uno supplies that and the is a common frequency utilized for IR modulation in the industry for signal detection.
- Different strategies were considered to get the robot from the laboratory walls to the inside of the laboratory. One strategy we thought of was making use of an Arduino compass to know the correct orientation of the robot to enter the lab. This was replaced by a more precise navigation strategy of making different sides of the lab with different colors and following an algorithmic flowchart accordingly.
- It was initially considered to use a continuous infrared signal for the emitter on the lab tower. A better alternative solution was made, which was the use of a high-frequency PWM signal in order for the robot to better differentiate between the blinking infrared light of the beacon and infrared light from the sun.

3 Component Description

3.1 Infrared Detection

The infrared sensor consists of an LED, a resistor, and a photo-transistor. While the LED emits infrared light, the photo-transistor receives the reflected light and changes its current which sequentially changes the output voltage, depending on the intensity of the reflected rays. The darker the color of the tape, the more rays will be absorbed and respectively, the brighter the color, the more rays will be reflected back.

The infrared sensors will be used to detect the boundary of the given area and the cliffs, which are marked by black tape. The infrared sensors will also be used to detect the rock samples, which are wooden blocks covered by white tape because by looking at the reflected rays, it is possible to identify the color of the surface. The analog signal of the infrared sensors will be the input to the Arduino.

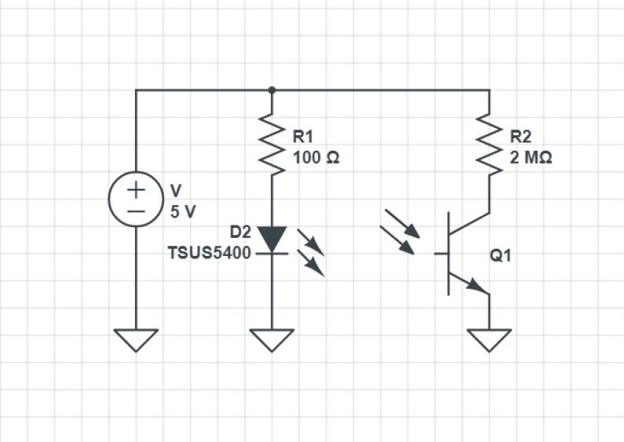


Figure 3.1: Circuit design of the IR sensor

3.2 Encoders and Gripper

Another important thing the robot should be able to do before it is able to navigate over the rough terrain of Venus is to have the capability to drive around. To achieve this, two encoders are installed at the front of the robot. Each encoder independently drives one big wheel, mounted at the front right and left side of the robot, making the robot able to drive forwards, backward, and spin around its own vertical axes. Combined with this, the robot also has one small spherical wheel at the back for needed stability.

The robot has four encoders in total, of which; two in the front to drive the big wheels, one in the back to operate the gripper, and one on the top to regulate the ultrasonic sensor's orientation. The gripper is important to achieve the goal of our mission; To explore the planet Venus and navigate through a maze of steep hills and craters and to find three hidden rock samples and bring them back to the predefined lab. The gripper will be necessary to pick up the rock samples and bring them back to the lab safely. The encoder in the back makes the gripper, which is installed at the front, move up and shut or down and open. This way the robot can drive over the rough terrain of Venus without the hindrance of the gripper, and can securely transport the rock samples.

The last encoder is responsible for making the ultrasonic sensors able to look around without turning the whole robot. With the inclusion of this extra feature, the ultrasonic sensors can scan a bigger area on the go, improving the efficiency of the mission. Helping the robot detect the mountains better so it can redirect its course before it drives into them.

All these encoders are controlled by an Arduino, located on the top of the robot. Functions for the encoders will be created, making them move a certain degree depending on the information received from the sensors; making the robot autonomous.

3.3 Ultrasound Sensor

The ultrasound sensor will be used to detect obstacles, such as mountains and walls, by calculating the distance between the robot and the object, based on the time required for the sound emitted by the sensor to reach the obstacle, reflect, and return back to the sensor. If a set threshold of a couple of centimeters is exceeded then the obstacle is close to the robot and movement in that direction is no longer possible.

The sensor itself is fixed in such a way that is able to receive accurate feedback from the surrounding area of the robot, the ultrasound sensor is attached to a servo motor, which enables the sensor to rotate 360 degrees.

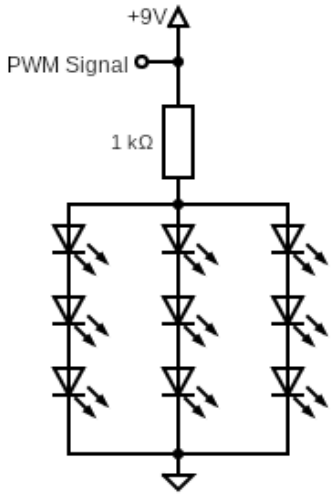
Based on the detection output and the direction from which the obstacle was detected, the ultrasound sensor, as well as the servo motor which is attached to it, will act accordingly in order to allow the robot to turn and change the direction of movement.

3.4 Laboratory Beacon

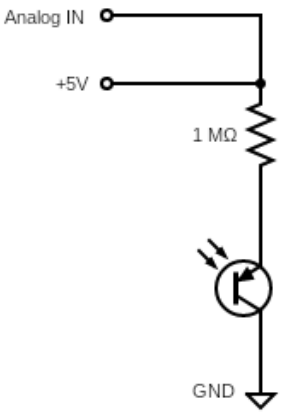
The objective of the following sub-module is to get the robot back to the lab after picking up the block. The robot will have a mounted infrared sensor as a tower, which will be used to detect the PWM signal of infrared light from the laboratory beacon. The robot will rotate until the lab is detected, and then drive forward, toward the lab. The laboratory walls will be detected using the closer-range infrared front sensors and the ultrasound sensor. The robot will be able to know which side it is on by putting different colors on each wall of the lab. At the walls of the laboratory, the robot will have a predefined flowchart-based set of actions based on the input from the infrared sensors.

The circuitry as showcased below will be utilized. For the Laboratory Tower. The Lab Beacon will be a construct of 3x3 sides of IR emitters, connected to an external 9V battery power source and a PWM signal input. We established that the 38kHz will be the best frequency, as it is a common frequency utilized for IR modulation in industry and will allow better detection of the signal.

The tower on the robot will contain an IR receiver connected to the Analog Input of the Arduino to identify the Beacon signal and calculate distance approximation. To estimate the distance we will be using the inverse square law. This law states that the intensity of light (or any other form of radiation) decreases with the square of the distance. If you know the intensity of the IR LED at a certain distance, you can compare it with the intensity measured by the IR photodiode and estimate the distance.



(a) Circuit diagram of the Beacon.



(b) Circuit diagram of the robot-mounted towers.

Figure 3.2: Circuit diagrams of the laboratory setup.

3.5 Communication

Communication between the robots can be implemented through the provided Zigbee modules on top of the robots. To overcome the challenges of this mission, communication is not necessarily needed but can be added to improve and optimize the initial solution. Because of this, communication will be left out for now.

The goal of the communication sub-module is to facilitate data sharing and effective collaboration between the two robots. The robots will carry out various tasks independently of one another, such as avoiding ramp usage overlap and picking up the same sample. The robots will communicate wirelessly using the Zigbee module to exchange data.

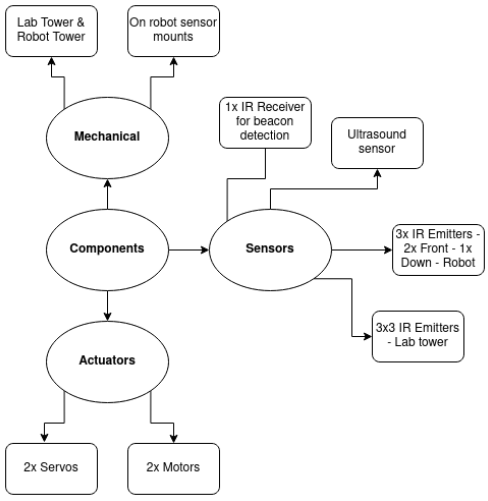


Figure 3.3: Systematic component overview of the entire robot/system.

4 Test Planning

4.1 Ultrasound Testing

- Test detecting capabilities of the sensor to determine whether the servo motor is needed. The test is going to take place by simulating some situations in which the robot moves toward the hill in a direction causing it to get stuck on the edges. If the sensor manages to detect the hill before the robot gets stuck then a servo motor is not strictly needed. A servo motor must be implemented if the robot does not detect the hill.
- Determine an optimal threshold distance to consider the obstacle and send information to the controller. The test will mostly consist in creating various situations where the robot interacts with obstacles that simulate the hills. The expected outcome is that the sensor will detect the hill and will communicate it before the robot hits it.

4.2 Infrared Testing

- Test the sensor to determine if the sensitivity is good enough to differentiate between the three cases (black and white tape and the ground). To test this, we can make separate measurements of each element and compare the values obtained. If there is a significant difference between the values, then the sensitivity is appropriate for the task.
- The software function needs to be tested. The expected outcome is that it gives back an integer depending on the output voltage. We also need to determine optimal cutoff values for when it detects the black tape and white tape. This test will be done with a trial and error method to narrow down the cutoff value.

4.3 Laboratory Testing

- Test detecting capabilities of the sensor to detect the infrared red light emitted from the beacon tower in the laboratory. This test can be done initially by testing if the sensor can see different infrared lights when a PWM signal is inputted to the emitter. The following feature will then be tested by simulating some situations from different distances, and seeing if the robot can detect the lab infrared light and move towards it. A big challenge for this test would be to see if the PWM signal we created at the beacon will allow the robot to differentiate between infrared light from the sun.
- Test the integration of our system with the infrared detection sub-module for the detection of the color of the laboratory walls. This testing will require setting different thresholds for different values of infrared radiation values received from different colors, and observing if it follows the algorithmic flowchart made. This will also be done by simulating situations and observing if it turns in the correct direction.

4.4 Movement Testing

Testing the movement of the robots is straightforward and does not require that much time. A few functions need to be tested to see whether the robot acts accordingly. These are:

- Software function to test whether the robot drives in a straight line (calibrate the wheel encoders). The expected outcome is that the robot indeed drives in a straight line.
- Software function to turn the robot to specifically given angles. Test and validate until the robot has an accuracy that is usable for the defined strategy. The expected outcome is that the robots will be able to turn roughly the right amount, but are not extremely accurate. This might require an additional solution if the accuracy is too poor.

4.5 Final integration

The final integration will require a lot of testing and optimizing as compared to the sub-module testing. The sub-modules are expected to be finished in week 5, according to the project planning in section 5.1. Specific testing has to be performed on the integrated software.
Straightforward, when the code is fully integrated, testing will be on a trial-and-error basis. Each module is first tested individually and the final code will be structured in a modular way, which allows for clear testing, debugging, and optimizing. When errors occur, the mistake can be easily connected to the specific sub-module. Testing will be done from the start of the strategy and work from there. Errors are expected, which is the reason final integration testing will be performed relatively early in the planning.

5 Project Planning

5.1 Group planning

Weeks	Wednesday	Friday	Deadlines
Week 3 (10 & 12 May)	Start working on separate sub-modules	Continue working on separate-sub-modules	12th May 11:59pm Design Report (15%)
Week 4 (17 & 19 May)	Continue working on separate-sub-modules	**NO MEETING**	-
Week 5 (24 & 26 May)	Testing of sub-modules	Finishing up the separate sub-modules	-
Week 6 (31 May & 2 June)	Integration of sub-modules	Integration of sub-modules	-
Week 7 (7 & 9 June)	Validation of software	Validation of whole system	-
Week 8 (14 & 16 June)	Optimizing the system/Start on video	Optimizing the system	-
Week 9 (21 & 23 June)	Finalize final report and video presentation	Review/discuss final report and video presentation	25th June 11:59pm Final Report (70%) & Video Presentation (15%)

5.2 Module planning

5.2.1 Mounts and Movement (Steven, Martijn & Bastiaan)

- Mount in front of the robot for infrared sensors (so the robot can see mountains and rock samples)
- Big tower-like mount on top of the robot for infrared sensors (to detect the lab)
- Functions for the driving of the robot
- A function for the gripper
- mount for the infrared beacon on top of the lab

5.2.2 Infrared Sensors (Ioanna, Adam & Jiahui)

- Design the circuit in order to detect the boundaries and the rock samples
- Make the code for the infrared sensor so as to differentiate between the black tape, normal ground, and white tape

5.2.3 Ultrasound Sensor (Stefano & Ana)

- Determine optimal threshold to detect an obstacle (in ms)
- Optimize code for detecting obstacles with the ultrasound sensor to filter out outliers
- Write a function for ultrasound servo motor (optional)
- Integrate sub-module in the system

5.2.4 Finding the laboratory (Malik & Daniel)

- Infrared distance detection circuitry
- Laboratory and robot beacon integration
- Algorithm to navigate into the entrance when it reaches the laboratory based on surrounding input