# DESIGN REPORT


**<u>Group 32</u>**

**Tutor:**

Avan D. B.

**Students:**

Steinebach L. ()

Stek T. (0957900)

Støle M. B. (0980907)

Stolle K. H. W. (0945431)

Timmerman N. A. G. (0962892)

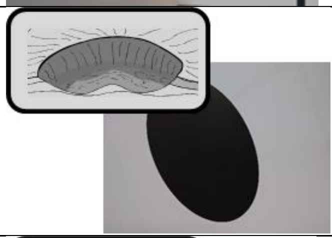
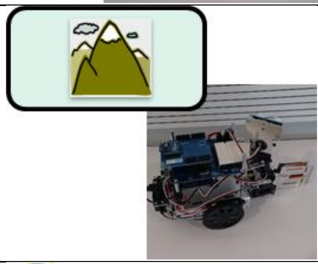
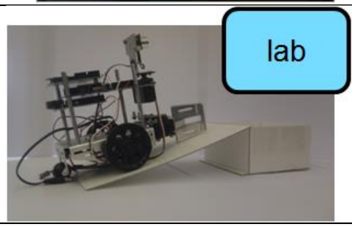
# Contents


System level description .... 4
- Problem description .... 4
- Strategy .... 5
- The robot .... 5
- Detailed specification of the components .... 6
- Microcontroller .... 6
- Actuators .... 7
  - Wheels .... 7
  - Servos .... 7
  - IR-LEDs .... 7
- Sensors .... 7
  - IR-sensor .... 7
  - Ultrasonic-sensor .... 7
- Test and integration plan .... 8
- Purpose .... 8
- Boundary and cliff sensor .... 8
- Identifying rocks .... 8
- Grabbing the rocks .... 8
- Mountain detection .... 8
- Communication .... 9
- Navigation .... 9
- Integration and final test .... 9
- Planning .... 9

# System level description

## Problem description

Our group was asked to design and program two robots, using a number of provided materials, which can execute a number of research tasks on Venus. The robots need to drive around on the planet and find as much research samples as possible in minimal time, while avoiding natural obstacles such as cliffs and hills. To simplify this assignment, a very abstract material model of Venus was made which will be used to test the behaviour of the robots. The model is summarized in the table below:

| **Object** | **Known properties** | |
|---|---|---|
| *Boundary*<br><br>Black tape | Absorbs infrared light. |  |
| *Cliff*<br><br>Black tape | Absorbs infrared light. |  |
| *Hill*<br><br>30cm high | Reflects ultrasound.<br>Absorbs infrared light. |  |
| *Rock sample*<br><br>White cardboard<br>2x2x2cm | Reflects infrared light. |  |
| *Lab*<br><br>Box with ramp<br>2.5cm high<br>20x20cm floor area | Reflects ultrasound. |  |

Translating the problem statement in terms of this material model, the robots need to find and pick up the rock samples and drop these in the research lab while avoiding the hills, cliffs and boundary. The robots have to communicate with each other in order to complete this task as efficiently as possible.

# Strategy

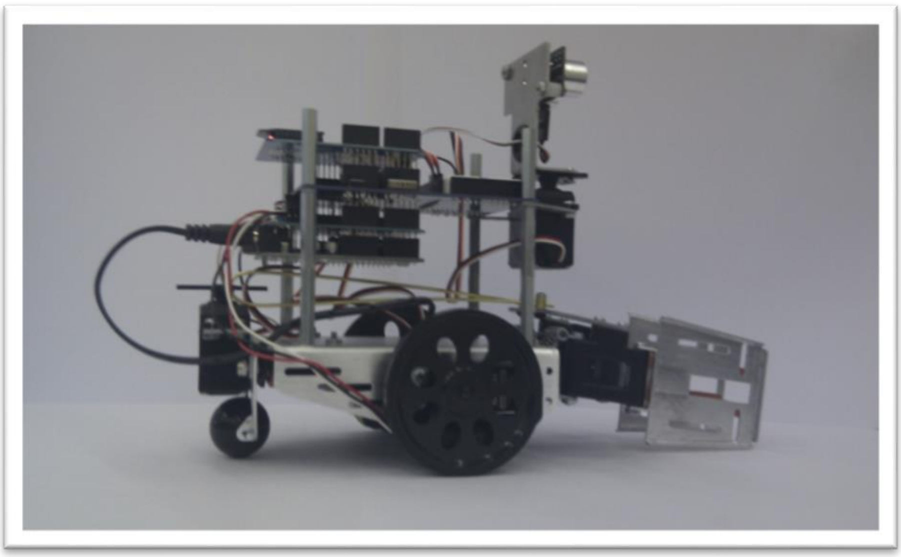
There are three relevant strategies of cooperation with the two vehicles.

1. Letting both of the vehicles simultaneously search the area for the rock samples and if a vehicle finds a tube it brings the rock to the lab. On this manner all the tube that are found are brought to the lab.
2. One of the vehicles is going to search for the tubes in the research area and drops the tubes at a fixed place in the field. The other vehicle waits until there is a tube at that fixed location and when there is, it brings the rock to the lab. Instead of letting the second vehicle wait in the beginning, it could also explore the area and also search for rock samples until there is found one rock sample.
3. Instruct one of the vehicles only to explore the area and communicate the locations of the rocks to the other vehicle. The other vehicle could than collet the tubes and bring them to the lab.

In order to know which strategy is the most efficient, or fastest, there should be performed some tests. With these tests we should investigate how much time certain jobs takes.

# The robot

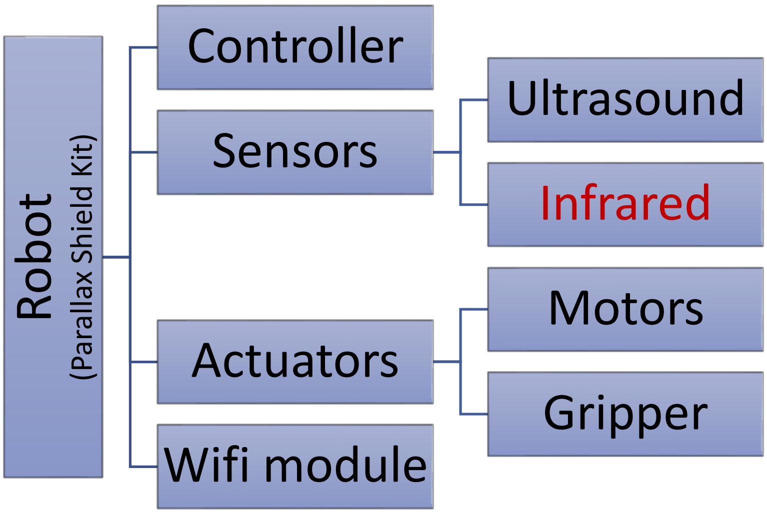
The provided robot is based on the 'Parallax Shield Kit', which contains a basic metal construction, an Arduino controller, a gripper and powered wheels. On top of this, the robot has an ultrasonic distance sensor, an additional breadboard and a wireless communication module.



In order to be able to succesfully execute the strategy described above, the robot needs to fulfill the following requirements:

- It can drive around.
- It can pick up and drop research samples.
- It can communicate with other robots for efficiency.
- It has a mechanism to detect hills, cliffs, research samples, the research lab and the boundary of the given area.

The first, second and third conditions are easily fulfilled, since the provided ‘bare’ robot contains powered wheels, a gripper, and a communication module. For the last condition though, an additional sensor may be needed. The hills can easily be detected by the ultrasound distance sensor. This cannot be used for the cliffs, boundaries and samples though, because these do not have a sufficient height. The only known property of these objects which can be used to detect and distinguish them is the infrared reflectivity. Therefore it was decided to add multiple infrared reflection sensors to the robot (marked in red in the picture, to distinguish it from the components included in the basic kit). These sensors can also be used to find the research lab. The robot and its core components are summarized in the graph below:




A more detailed description of these core components and their positions will be given in the next section.

# Detailed specification of the components

## Microcontroller

The Venus exploration robot has an Arduino Uno (ATMega38P-based) microcontroller at its core.

Arduino Uno runs at an operating voltage of 5V, has 14 digital I/O pins, 6 PWM-digital I/O pins, 6 analog input pins and uses the AVR-architecture.

The Arduino Uno can be programmed using the “Arduino Language”, a language that is ‘transpiled’ to C and then compiled using ‘avr-gcc’ (a version of ‘gcc’ specifically for the AVR-architecture). The language is essentially a subset of C that hides some of the less intuitive parts of the AVR architecture from the programmer. Arduino Uno does not allow for concurrent programming, but does have a system in place for software interrupts, allowing the programmer to schedule periodic tasks (e.g. checking the input of a sensor).

# Actuators

## Wheels

The robot has two wheels; each wheel is driven by a separate electronic motor.

The speed at which a wheel turns can be modulated using a PWM-signal. Below the maximum speed, it is not possible to directly control the speed at which the wheel is turning. This would require some feedback or tracking system. The signal sent to the wheel will only control its acceleration.

## Servos

The robot has two servos, one to control the grabbing of objects and one to control the angle of the ultrasonic sensor on top of the robot. A servo is an electronic motor that uses a servomechanism to provide itself with negative feedback, allowing for very precise control of the angle made by the motor mechanism.

This angle is dictated by a PWM-signal.

## IR-LEDs

An IR-LED (Infra-Red Light Emitting Diodes) will emit infrared light when a current is flowing through it. IR-LEDs act much like regular diodes.

# Sensors

## IR-sensor

Infra-red light is a type of light with a frequency close to red light, but outside of the visible spectrum. An IR-sensor will measure the intensity of red light – from all directions – cast upon it.

The effective range (i.e. the range from which IR-light may be emitted for a significant measurement) is 100cm for detection purposes and 20cm for analysis purposes.

## Ultrasonic-sensor

An ultrasonic sensor (Parallax 28015 REV C) will check the time it takes for a self-emitted ultrasound wave to reflect back to itself. This allows for precise calculation of distance.

The difference between the point in time when a sound was emitted and the point in time when the sound was reflected and detected allows for calculation of the distance the sound has travelled.

It should be noted that the ultrasonic sensor on the Venus explorer robots is located on top of the robot, and there is no possibility of tilting it in a downwards angle. This means that the robot can only be used to detect objects that are tall enough to ‘catch’ the ultrasound emitted by the sensor.

The range for this specific sensor is 2cm to 300cm.

# Test and integration plan

## Purpose

The purpose of the test and integration plan is to describe the necessary tests to verify that all the modules developed for this OGO Venus Exploration project works in a proper way.

## Boundary and cliff sensor

To test this subsystem we will use a sheet of paper or table (white) with black lines and circles in order to simulated the Venus area. The robot will then be put inside the boundaries at a random position and turned on to test if it avoids the black parts. This test will help to fine-tune the code and position of the sensors so everything can run smoothly.

## Identifying rocks

To test if the robot can detect all the rocks in the search area, a route will be made with rocks at different locations. The robot will then drive this route and count the rocks. This will be done several times with the rocks at different locations. When the robot is successful in locating the rocks, the identifying of distance from a rock to the robot will be tested. This can be done in three steps starting with a basic test were the robot needs to calculate the distance from a standstill position with a rock placed in front of it. The second step would be to the same test but with a rock placed to the side so the robot needs to turn in order to have the rock right in front of its sensors. The last step would be to do test with all the rocks in the area at the same time.

## Grabbing the rocks

The test to make sure that the robot can grab the rocks will be conducted in the same way as the test for locating the rocks with three different steps. First is to grab a rock located in front of the robot, the second and third test will be to grab a rock located to the left and to the right of the robot. If these tests are successful we can be certain that the robot will manage to grab any rock in the search area.

## Mountain detection

In order to test the system for detecting mountains, the robot will have to perform several movements whenever it closes in on a mountain. By using serial communication showing the distance sensed by the robot and the true distance, the accuracy and any errors can be measured. Testing of the servos needs to be conducted in order to check if the robot uses them to look left and right and giving accurate numbers. As a final test, the robot need to move around a mountain by itself.

### Communication
The communication part will only be implemented after the robot can identify and grab rocks, and avoid cliffs and mountains. The first part to be implemented is the communication between the two robots after they have grabbed a rock and are bringing it back to the lab. Only one robot can do this at one certain time. This can be tested by giving both robots a rock at the same time. The robot who gets the rock first should move to the lab, while the other one should wait.

### Navigation
In this test a copy of the possible surface used in the final test will be used. The first test will only involve one of the two robots in order to see that it behaves as expected, which basically means that it need to move in the correct directions and return to the lab with the rocks. If the first robot succeeds, then the second robot will conduct the same test. If both of them perform as expected, the test will be conducted again, but this time with both robots placed in the testing area. These tests should also give an indication whether or not the robots follow the lines it is supposed to, if they notice any rocks they have already picked up and if they manage to get back to the lab with the rocks. If there is enough time, this test should be split up into several smaller test to check each aspect of the navigation.

### Integration and final test
Once all subsystems are performing as planned (independently), all the systems will be combined. Then a test to check that everything works together as planned is to be conducted. There might be some unforeseen complications when all the different parts and codes are combined. Some extra time should be dedicated to the final integration in order to solve these potential complications.

## Planning

| **When** | **What** | **Extra Information** |
|---|---|---|
| Week 1 | Task division.<br>Global setup.<br>Strategy Choice. | Deadline Design report 1-5-2016 |
| Week 2 | Programming basic Utilitarian functions.<br>Examining robot functionality<br>Writing Design Report | Ordering extra components |
| Week 3 | Testing Basic functionality Robots.<br>Developing main Algorithm | |

| | | |
|---|---|---|
| | Start of writing Final Report<br>Making a task division | |
| Week 4 | Continuing development of strategy<br>Developing main Algorithm<br>Implementing and programming of extra components<br>Begin testing fase | Expected delivery of extra components |
| Week 5 | Testing functionality and sensor accuracy<br>Making of video presentation | |
| Week 6 | Testing functionality, sensor accuracy and speed sufficiency<br>Making of video presentation | |
| Week 7 | Testing functionality and sensor accuracy<br>Finishing video presentation<br>Finalizing Complete project | Deadline video presentation: 16-06-2016<br>Deadline Final Report:19-06-2016<br>Demonstration:17-06-2016 |