

NOT MARS ROVER

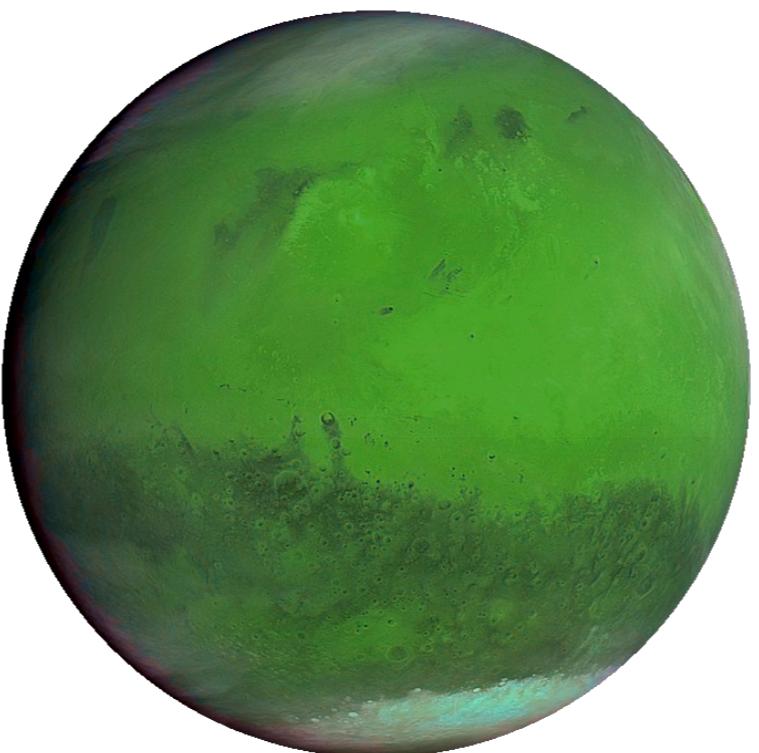


Figure 1: Not Mars (Sturges, 2025)

Contents

| | | |
|--------------------|--|----|
| 1 | Introduction | 2 |
| 1.1 | Goals | 2 |
| 1.2 | Assumptions | 2 |
| 1.3 | Tools Used | 2 |
| 2 | Identifying Aspects a Solution Will Have | 2 |
| 2.1 | Success Criteria | 2 |
| 2.2 | Evaluation | 3 |
| 2.3 | Existing Designs | 3 |
| 2.3.1 | Perseverance | 3 |
| 2.3.2 | Spirit & Opportunity | 4 |
| 2.3.3 | Zhurong | 4 |
| 3 | Logistics | 4 |
| 4 | Design | 4 |
| 4.1 | Initial Designs | 4 |
| 4.2 | Revised Design | 4 |
| 4.3 | Sensors | 5 |
| 4.4 | Gearing | 5 |
| 4.5 | Software Architecture | 5 |
| 4.5.1 | Language Choice | 5 |
| 4.5.2 | Design | 5 |
| 4.5.3 | Development | 6 |
| 5 | Design Review | 6 |
| 5.1 | Materials Selection | 7 |
| 5.2 | Inclined Plane | 7 |
| 5.3 | Friction | 7 |
| 5.4 | Gear Ratios | 8 |
| 6 | Testing | 8 |
| 7 | Conclusion | 8 |
| 7.1 | Recommendations | 8 |
| 7.2 | Limitations | 8 |
| Bibliography | | 8 |
| Appendix | | 10 |
| Source Code | | 10 |

1 Introduction

1.1 Goals

The goal of this report is to document the creation of a prototype for an extra-planetary rover, similar to those that are sent to mars and the moon. This report will also evaluate the success of the design, and the logistics of topics surrounding the design.

1.2 Assumptions

1. The rover will be going to an earth-like, terrestrial, planet not inside of our own solar system, this planet will henceforth be referred to as "not mars"
 - Task requirement
2. The rover prototype will be made from LEGO technic pieces with an LEGO EV3 as the controller and power source
 - Task requirement
3. Only one driver motor is allowed on the rover
 - Task requirement
4. The rover is at a 1:10 scale
5. The rover's main driver motor will be 25kw at full scale
 - Task requirement
6. The rover will need to carry 250kg of payload (full scale) up a 30° incline.

1.3 Tools Used

Several tools were used in the project, including design, programming, and document creation.

- Typst
 - Document typesetting language
 - <https://github.com/typst/typst>
- Gimp
 - Used to create and edit images
 - <https://www.gimp.org/>
- Fish shell
 - Used for various automations
 - <https://fishshell.com/>
- Rust programming language
 - The rover was programmed entirely in rust
 - <https://www.rust-lang.org/>
- GNU make
 - GNU make was used to automate the compilation, deployment, and running of the rover software
 - <https://www.gnu.org/software/make/>
- Docker

- Docker was used to create lightweight virtual machines in order to cross compile code for the EV3's CPU
 - <https://www.docker.com/>
- ev3dev
 - Linux operating system designed to run on the LEGO EV3 bricks
 - <https://www.ev3dev.org/>
- ev3dev-lang-rust
 - Rust language bindings for ev3
 - <https://github.com/pixix4/ev3dev-lang-rust>
- OpenSSH
 - OpenSSH is an open source framework for making secure shell (SSH) connections, these are used to remotely configure and test the EV3, as well as deploy the code to it.
 - <https://www.openssh.com/>
- Helix
 - Postmodern modal text editor
 - <https://helix-editor.com/>
- rust-analyzer
 - LSP and code analyzer for rust
 - <https://rust-analyzer.github.io/>
- Nix
 - Declarative development shells and package management
 - <https://nixos.org/>
- NixOS
 - Linux operating system based on the nix package manager
 - <https://nixos.org/>
- Git
 - Version management tool used to manage the source code of the project, as well as the source of this document
 - <https://git-scm.com/>
- GitHub
 - Used to back up the source code of this document and the software, and allow for collaboration
 - <https://github.com>
- Fletcher
 - Used to create finite state machine diagrams and flow charts
 - <https://typst.app/universe/package/fletcher/>

2 Identifying Aspects a Solution Will Have

2.1 Success Criteria

A successful prototype will be able to ascend a 35° inclined plane made of several materials such as gravel, AstroTurf, and sand. The rover will also stop if the incline is greater than or equal to 40°, and if it detects any obstructions in front of it. The rover in this prototyping

phase does not require the ability to steer or any advanced control logic, as the goal is to prototype the drivetrain.

2.2 Evaluation

A successful prototype will be evaluated though testing on a ramp at a 35° angle consisting of sand, gravel, and AstroTurf. This ramp is designed to simulate the various materials the ground could consist of on Mars, as zero missions have been sent there before to discover the surface qualities, so the rover needs to be adaptable.

2.3 Existing Designs

In this section existing rover designs will be analysed, including their design, flaws, and successes.

2.3.1 Perseverance

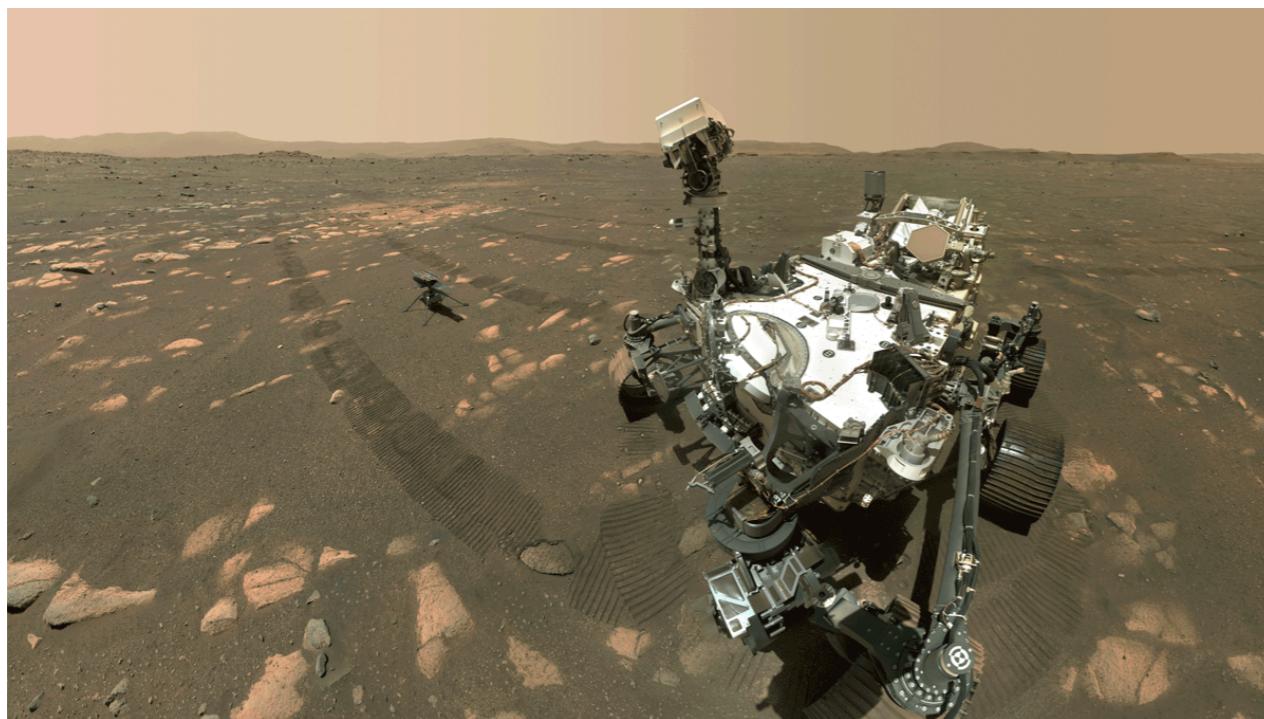


Figure 2: Perseverance with Ingenuity (NASA, n.d.-a)

The perseverance rover was launched atop an Atlas-V rocket on 2025-07-19 11:50:00 UTC (NASA, n.d.-b). It landed on mars on 2021-02-18 (Landers, 2021). The *Perseverance* rover shares a very similar appearance and design with its predecessor, *Curiosity*. The design was improved upon based on data gathered from the *Curiosity* rover's time on mars.

Perseverance also took a small helicopter with it.

An example of the improvements of the *Perseverance* rover is the wheels. On sol 411 of the *curiosity* rover's mission a rip was noticed in one of the wheels. *Perseverance* improved on this by making the wheels thicker and slightly larger (Lakdawalla, 2014).

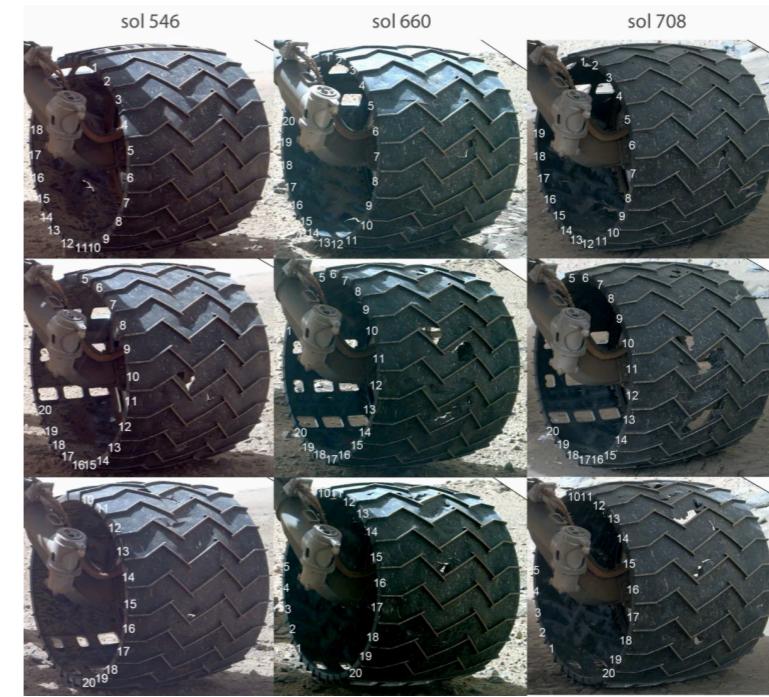


Figure 3: The *Curiosity* rover's wheel damage (Lakdawalla, 2014)

Perseverance contains drive motors in each individual wheel, and steering motors on both the back and front wheels, but the middle wheels are fixed (NASA, n.d.-c). This is not applicable to our design, as the EV3 does not have enough output ports to drive 10 motors, and only one drive motor is allowed for the prototype.

2.3.2 Spirit & Opportunity

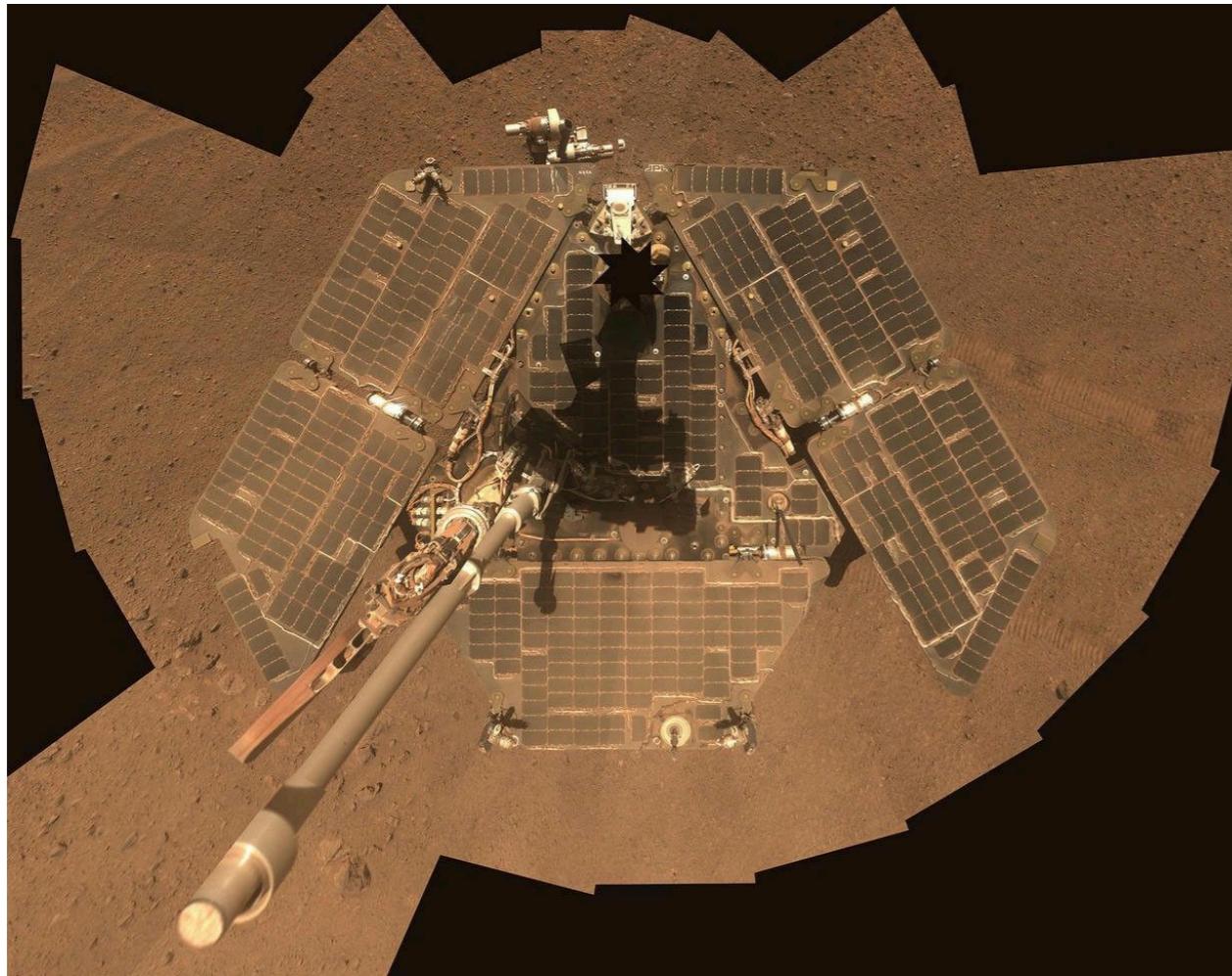


Figure 4: Opportunity as seen from above (NASA, 2019)

Spirit and Opportunity were 2 identical rovers that touched down on mars on 2004-01-03 and 2004-01-24 respectively. Their goal was to find evidence of water having existed on mars in the past (NASA, 2019).

Opportunity, like perseverance also uses individual motors on each wheel.

2.3.3 Zhurong

Zhurong is a Chinese rover launched on 2020-08-23 and landed on mars on 2021-05-14 (Myers & Chang, 2021). It's mission ended on 2022-05-20 when dust covered the solar panels. It also has motors individually powering each wheel, like Spirit, Opportunity, and perseverance.

3 Logistics

The task specifies that the rover will weigh 250 kg in total. For simplify the interstellar transfer module will be assumed to be a separate launch and all that needs to be done is to

get the 250kg rover to low earth orbit. According to a calculator provided by SpaceX (n.d.), a launch of 250kg to low earth orbit would cost \$1.64 M USD.

4 Design

4.1 Initial Designs

Initial concepts were brainstormed on a whiteboard, the first designs involved having the bottom and top of the robot separate, with a arm between them that can hold sensors, the arm was supposed to have 2 joints, and the 'head' would stay level on the ground to ensure stable sensor data, and to see further than it could if the sensors were near the ground. This idea stuck around the entire time until the end.

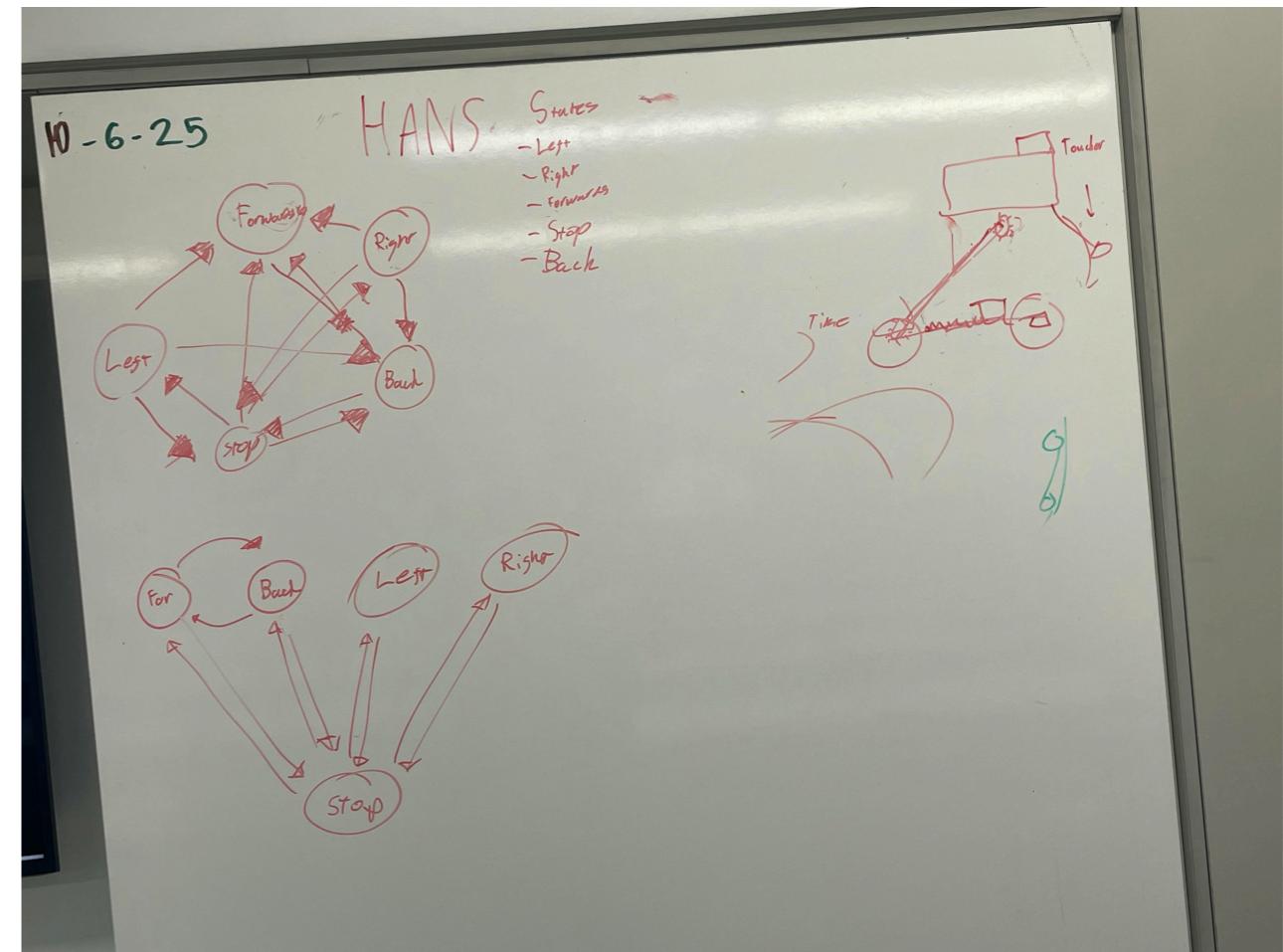


Figure 5: Whiteboard that the initial designs were drawn on

4.2 Revised Design

In the later stages of building the rover it became clear that making the swiveling arm was not possible, first, the lego pieces could not put up with the mechanical stress, second the lego motors could not hold up the head. We were also running out of time. We revised the design to be a more simple design that could only go forwards and backwards, however with the added simplicity we were able to make it four wheel drive.

Later we added on more wheels and moved the motor to ensure correct gearing.

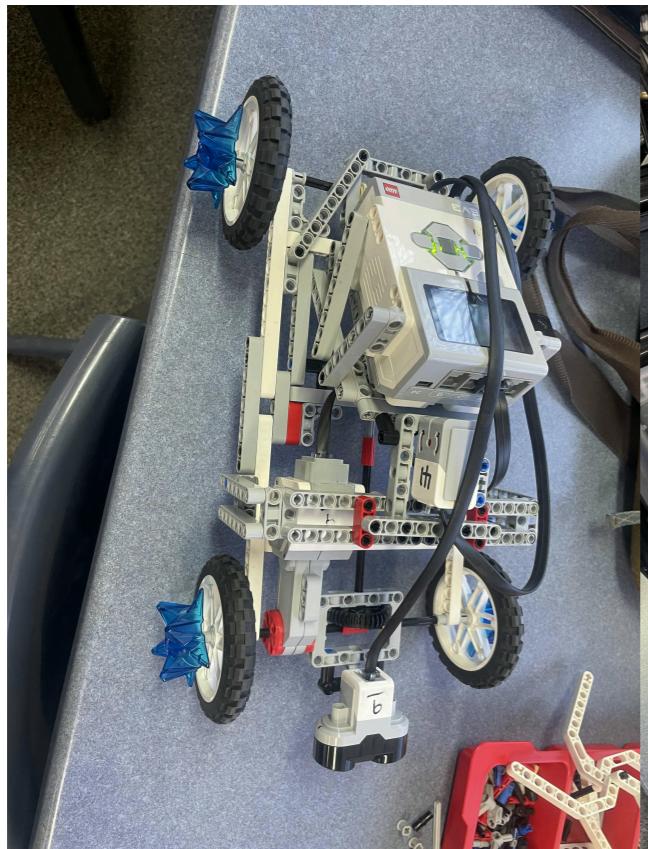


Figure 6: Revised design.

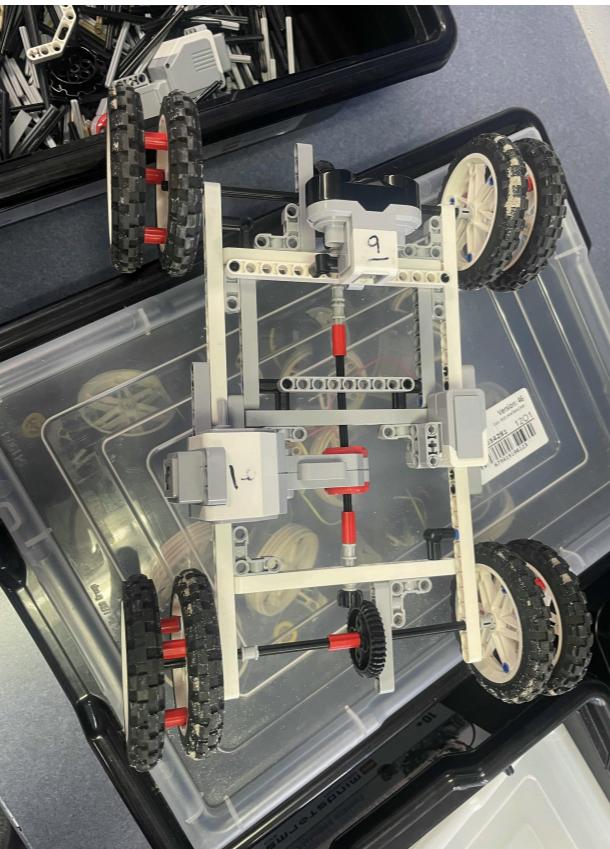


Figure 7: Revised design with extra wheels and moved motor.

4.3 Sensors

The original design was intended to have 3 sensors: an ultrasonic sensor, a touch sensor and a gyroscope. The touch sensor was removed from the design, because its usefulness was debatable, and it added more complexity to the design. The ultrasonic sensor is used to detect objects, and the gyroscope sensor to detect tilt.

4.4 Gearing

A low gear ratio was selected to regulate the speed of the rover, and to increase its ability to climb the inclined plane.

4.5 Software Architecture

4.5.1 Language Choice

By default, the EV3 can only be programmed with a block coding language, this is fine for simple things, and could be fine for the project, but personal preference and expandability lead me to using the rust programming language.

Rust is a low-level systems language which features memory safety through its borrow checker, making it very hard to induce memory related undefined behaviour. This contrasts

with comparable languages such as C++ and C where it is very easy to cause security and stability issues through memory bugs.

4.5.2 Design

The software was modeled with a finite state machine. A finite state machine is a program with a set amount (finite) of states, and defined ways of moving between states. For example, we could have a finite state machine that lets us move forwards, but only after we have stopped. This design was chosen because it is very simple and can be expanded. To add more states all that needs to be done is define the state and its transitions.

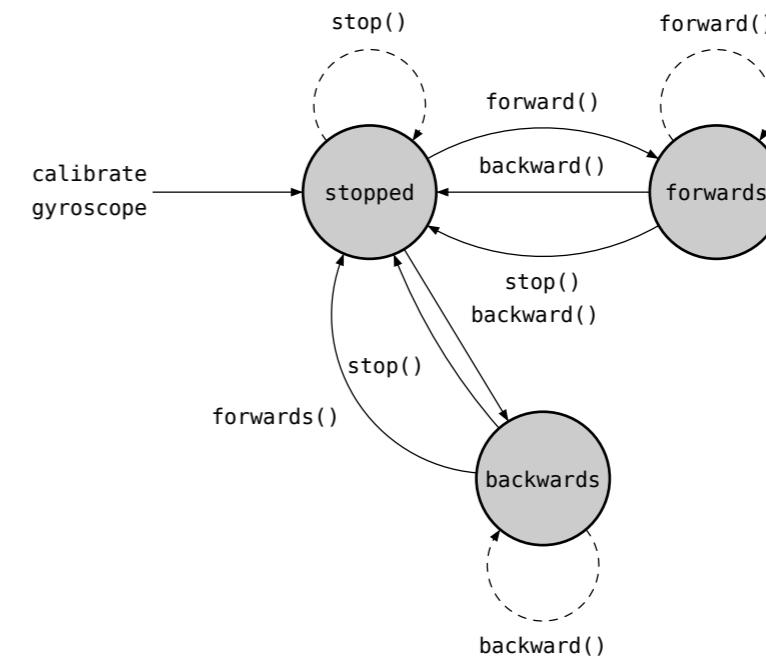


Figure 8: The finite state machine for the robot (Sturges, 2025)

However, these states do not fully encapsulate the logic of the rover, only the restrictions of moving between different movement states. Below is the full logic for when and why moving between states would be done.

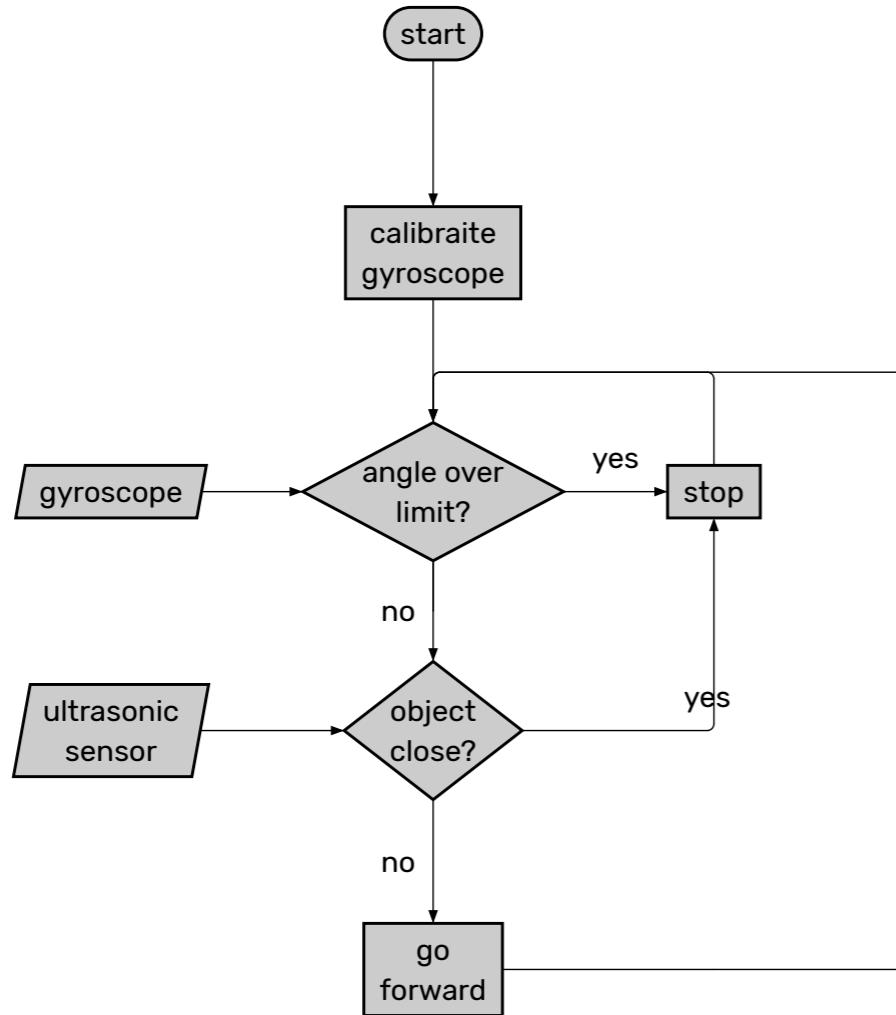


Figure 9: Flow chart of rover logic (Sturges, 2025)

4.5.3 Development

The codebase was split between two files, `main.rs` and `lib.rs`, as is standard in rust projects. `main.rs` contains logic for setup, and for reacting to sensor inputs while `lib.rs` contains struct definitions and the logic for moving between states.

```

1 #[derive(Default, PartialEq, Eq, Clone, Copy, Debug)]
2 pub enum MoveState {
3     #[default]
4     Stop,
5     Forwards,
6     Backwards,
7 }
8
9 #[derive(Default, Clone, Copy, Debug)]
10 pub struct RobotState {
11     move_state: MoveState,
12     sensor_data: SensorData,

```

```

13 }
14
15 #[derive(Default, Clone, Copy, Debug)]
16 struct SensorData {
17     angle: i32,
18     distance: i32,
19 }

```

The `RobotState` struct contains both the movement state and the `SensorData`, which is the processed sensor data. The `RobotState` struct has `stop()`, `forwards()` and `backwards()` methods implemented on it, which tell the motor to move in the correct direction, as well as changing the `MoveState` enum in the `RobotState`. The implementation of the motion can be seen in the appendix.

```

1 robot.setup(&mut peripherals)?;
2 peripherals.calibrate_gyroscope()?;
3 robot.forwards(&mut peripherals)?;
4
5 loop {
6     robot.update_sensor_data(&peripherals)?;
7
8     while robot.sensor_data.angle >= 35 {
9         robot.stop(&mut peripherals)?;
10    }
11
12    while robot.sensor_data.distance <= 10.0 {
13        robot.stop(&mut peripherals)?;
14    }
15
16    robot.forwards(&mut peripherals)?;
17 }

```

rust

The above section of the `main()` function (the main entry point for a binary program in rust) shows the entirety of the control logic. Most of the programs complexity is in the implementation of the above methods such as `.stop()` and `.forwards()`. That complexity is abstracted away to allow for simple customisation of logic as seen above. On startup, the robot's setup function is run, then the gyroscope is calibrated. We then begin the main loop where we first update the data coming in from the sensors. We then check if the angle is over 35 degrees, and if it is we order the robot to stop, then complete the loop early. If the robot was not over 35 degrees in tilt, we then check if the distance on the ultrasonic sensor is less than or equal to 10cm, and if it is we stop and complete the loop early. If neither above case was true the robot continues forwards.

5 Design Review

This section contains analysis of the prototype as well as information on things not included in the prototype such as materials selection.

5.1 Materials Selection

The conditions on Not Mars are completely unknown, they could range from a martian atmosphere (cold and thin) to one more similar to Venus (hot and high pressure). Therefore, materials need to be selected that have high melting points and can take a large amount of tensile load.

Venera 7, the first probe to ever land on Venus is a good case to examine, the probe was the first to go to Venus at a time when little was known about the surface, so it was massively overbuilt (Huntress & Ya Marov, 2011). This is likely what would need to be done for the case of the Not Mars rover, as nothing is known of the surface of Not Mars.

The pressure vessel of Venera 7 was built from a solid hollow sphere of titanium with no welds (Williams, n.d.). The Not Mars rover should do something similar, with a sphere or elongated sphere for the main body made from titanium to put up for the potentially high pressure and high temperatures. Titanium has a melting point of 1725°C (United Performance Metals, n.d.), while Venus has a surface temperature of only 475°C. This means that if Not Mars is closer to Venus than to earth, it should not instantly melt.

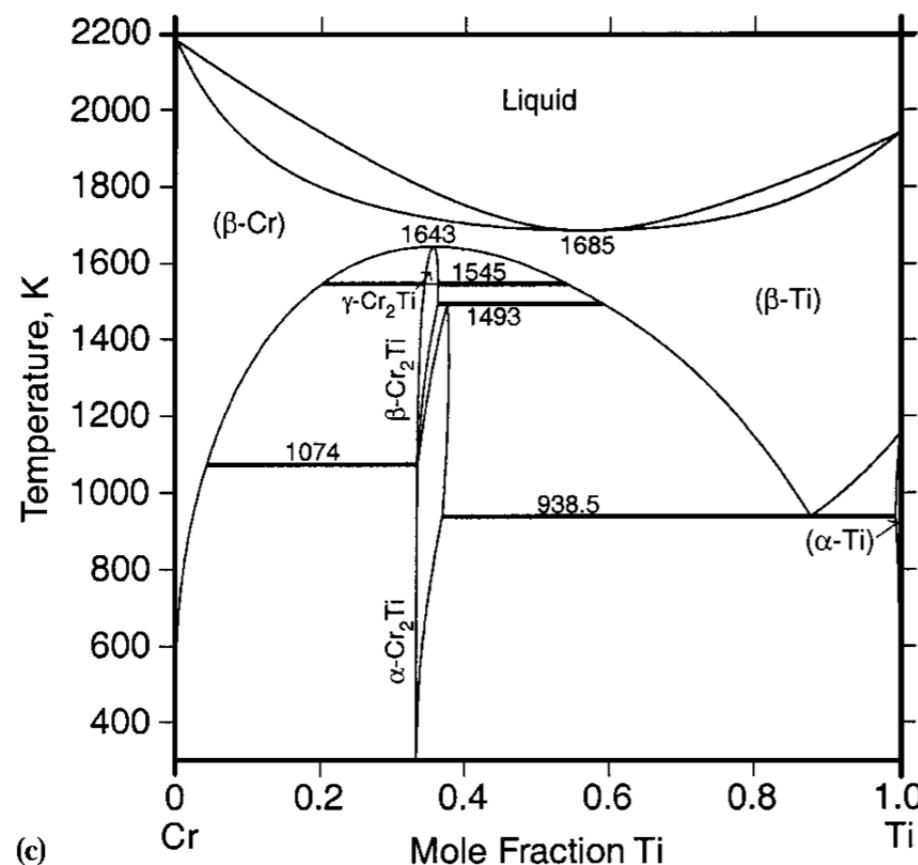
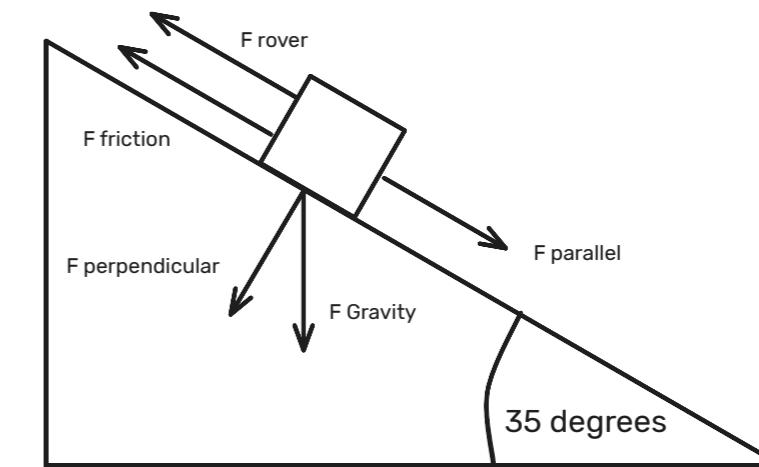


Figure 10: Phase diagram of titanium and chromium alloy

5.2 Inclined Plane

The mass of the rover was measured to be 685 grams, using this the forces on the rover on the inclined plane can be calculated.



$$\begin{aligned}\theta &= 40^\circ \\ F_g &= mg \\ F_g &= 6.713 \\ F_{\parallel} &= \sin(\theta) \cdot F_g \\ F_{\parallel} &= 4.315 \\ F_{\perp} &= \cos(\theta) \cdot F_g \\ F_{\perp} &= 5.142\end{aligned}$$

As can be seen in the above math, the parallel force from gravity is 4.315 N, meaning the rover going up the incline will need to be able to move itself with 4.315 N of force. Scaled up to the scale factor of 1:10 this is 43.15 N that the final Not Mars rover would need to overcome to ascend the 40 degree incline.

5.3 Friction

The testing ramp is made from three materials in the following order: sand, gravel, and AstroTurf. This is intended to provide a realistic simulation of what conditions could be expected on not mars.

| Material | Coefficient of Friction (μ) | Source |
|-----------|-----------------------------------|---------------------------|
| Sand | 0.55 | fine (n.d.) |
| Gravel | 0.6 | fine (n.d.) |
| AstroTurf | 0.35 | Victoria Goverment (n.d.) |

Using this information the force required considering friction can be calculated with the equation μF_{\perp} .

| Material | Force Required | Real Scale Force Required |
|-----------|----------------|---------------------------|
| Sand | 2.8281 N | 28.271 N |
| Gravel | 3.0852 N | 30.852 N |
| AstroTurf | 1.7997 N | 17.997 N |

5.4 Gear Ratios

The rover has a 1.8:1 gear ratio, this means that the rover will output 1.8 times more torque at the cost of 0.8 times the speed. This is actually beneficial, as the rover does not need to go fast, and the ability to overcome obstacles is more important than the speed. The perseverance rover has a top speed of 0.16 kph (NASA, n.d.-b), this is because the rover does not need speed, and speed is dangerous. The round trip delay between earth and mars is 27.8 minutes, this means that the time to send an instruction then see the results of that instruction is 27 minutes. This means that real time piloting of the rover is impossible, and instead of manual driving, you tell the rover to go this direction for this long.

6 Testing

Testing the prototype on the test ramp showed several design flaws:

1. The rover would get stuck in the sand, despite the extra wheels added, the wheels would dig into the sand and it would sink, putting too much strain on the drivetrain and leading to the damaging of parts.
2. The rover would struggle to go up the gravel section, as it would dislodge the gravel as it moved.
3. The rover did not have enough traction to make it up the AstroTurf section.

Although the rover suffered from several flaws, the software functioned as intended.

7 Conclusion

7.1 Recommendations

It is recommended that further research and prototyping is done, preferably with technologies such as 3d printing and other machining methods to provide more design control. The restriction to LEGO parts is limiting as parts cannot be designed for strength, as well as the parts being too big to provide a realistic size for the final scaled design. Further prototypes should also use low cost microcontrollers such as the ESP32, or even the Raspberry Pi. Using more advanced controllers such as these do not significantly add to the complexity of the software, but allow for more flexibility and design ideas to be evaluated, such as using machine learning to autonomously drive the rover and avoid obstacles. Using other microcontrollers also allows for more design freedom, as they will be lighter and smaller than the LEGO EV3, which is quite large and heavy. Components like batteries can also be separated further from the controller allowing the layout of electronics to conform to the design of the body rather than be in a large brick that needs to be designed around.

7.2 Limitations

The prototype was limited by the usage of LEGO, as it often failed to provide the required structural strength to deal with the forces applied by the rover's motor. The LEGO also had limitations in that the EV3 brick needed to be designed around.

Launch costs for the rover are nearly impossible to evaluate, as there are no comparable space missions. Humanity has not yet sent a rover interstellar, only the two voyager probes have ever been launched on an escape trajectory of the earth. The report had to assume that the payload would be launched to low earth orbit, where a transfer stage would later pick the payload up and transport it interstellar.

Bibliography

- Al Jazeera. (2023). *China breaks silence over status of Mars rover Zhurong*. <https://www.aljazeera.com/news/2023/4/25/china-breaks-silence-over-status-of-mars-rover-zhurong>
- fine. *Table of Ultimate Friction Factors for Dissimilar Materials*. Retrieved July 27, 2025, from <https://www.finesoftware.eu/help/geo5/en/table-of-ultimate-friction-factors-for-dissimilar-materials-01/>
- Huntress, W., & Ya Marov, M. (2011). *Soviet Robots in the Solar System: Mission Technologies and Discoveries*.
- Lakdawalla, E. (2014). *Curiosity wheel damage: The problem and solutions*. The Planetary Society. <https://www.planetary.org/articles/08190630-curiosity-wheel-damage>
- Landers, R. (2021). *It's landing day! What you need to know about Perseverance Rover's landing on Mars*. Florida Today. <https://www.floridatoday.com/story/news/local/2021/02/18/mars-perseverance-rover-land-red-planet-ingenuity/6765226002/>
- Myers, S., Lee, & Chang, K. (2021). *China's Mars Rover Mission Lands on the Red Planet*. New York Times. <https://archive.is/6HPMI>
- NASA. (n.d.-b). *Mars 2020: Perseverance Rover*. Retrieved July 19, 2025, from <https://science.nasa.gov/mission/mars-2020-perseverance/>
- NASA. (n.d.-c). *Mars 2020: Perseverance Rover Components*. Retrieved July 19, 2025, from <https://science.nasa.gov/mission/mars-2020-perseverance/rover-components/>
- NASA. (n.d.-a). *Perseverance's Selfie with Ingenuity*. Retrieved July 19, 2025, from <https://science.nasa.gov/resource/perseverances-selfie-with-ingenuity/>
- NASA. (n.d.-d). *Venus Facts*. Retrieved July 27, 2025, from <https://science.nasa.gov/venus/venus-facts/>
- NASA. (2019). *Mars Exploration Rovers: Spirit and Opportunity*. <https://science.nasa.gov/mission/mars-exploration-rovers-spirit-and-opportunity/>
- SpaceX. *Smallsat Rideshare Program*. Retrieved July 19, 2025, from <https://www.spacex.com/rideshare/>

United Performance Metals. *Titanium*. Retrieved July 27, 2025, from <https://www.upmet.com/products/titanium>

Victoria Goverment. *Artificial Grass For Sport*. Retrieved July 27, 2025, from https://sport.vic.gov.au/__data/assets/pdf_file/0005/2260238/Artificial-Grass-For-Sport-Guide-Part-3.pdf

Williams, D. *Venera 7*. Retrieved July 26, 2025, from [https://nssdc.gsfc.nasa.gov/nmc\(spacecraft/display.action?id=1970-060A](https://nssdc.gsfc.nasa.gov/nmc(spacecraft/display.action?id=1970-060A)

Appendix

Source Code

Source code is licensed under the AGPL-3 licence. The name of the file is in the top left of the code block.

File Structure:

```
1 ev3-rust
2 └── Cargo.lock
3 └── Cargo.toml
4 └── flake.lock
5 └── flake.nix
6 └── LICENSE
7 └── Makefile
8 └── src
9   └── lib.rs
10  └── main.rs
```

nix

ev3-rust/flake.nix

```
1 {
2   description = "A very basic flake";
3
4   inputs = {
5     nixpkgs.url = "github:nixos/nixpkgs?ref=nixos-unstable";
6
7     fenix = {
8       url = "github:nix-community/fenix";
9       inputs.nixpkgs.follows = "nixpkgs";
10    };
11  };
12
13  outputs = { self, nixpkgs, fenix };
14  let
15    system = "x86_64-linux";
16    pkgs = nixpkgs.legacyPackages.${system};
17    fe_pkgs = fenix.packages.${system};
18
19    librarys = with pkgs; [
20    ];
21  in {
22    devShells.${system}.default = pkgs.mkShell {
23      buildInputs = with pkgs; [
24        (fe_pkgs.complete.withComponents [
25          "cargo"
26          "clippy"
27          "rust-src"
28          "rustc"
29          "rustfmt"
30        ]
31      );
32      clang
33      lld
34      pkg-config
35      rustup
36      gnumake
37      cargo-cross
38      ] ++ librarys;
39
40      shellHook = ''
41        echo !!!!!!!!
42        echo
43        echo "docker pull pixix4/ev3dev-rust:latest"
44        echo
45        echo !!!!!!!!
46      '';
47
48      LD_LIBRARY_PATH = "${pkgs.lib.makeLibraryPath librarys}";
49    };
50  };
51 }
52
```

toml

ev3-rust/Cargo.toml

```
1 [package]
2 name = "ev3-rust"
3 version = "0.1.0"
4 edition = "2021"
5
6 [dependencies]
7 env_logger = "0.11.8"
8 ev3dev-lang-rust = { version="0.15", default-features=true, features=["ev3"] }
9 log = "0.4.27"
10
11 [profile.release]
12 lto = true
13 strip = true
14 opt-level = "s"
15
```

rust

ev3-rust/src/main.rs

```
1 // ev3-rust
2 // Copyright (C) 2025 skoove
3
4 // This program is free software: you can redistribute it and/or modify
5 // it under the terms of the GNU Affero General Public License as published by
6 // the Free Software Foundation, either version 3 of the License, or
7 // (at your option) any later version.
```

```

8
9 // This program is distributed in the hope that it will be useful,
10 // but WITHOUT ANY WARRANTY; without even the implied warranty of
11 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 // GNU Affero General Public License for more details.
13
14 // You should have received a copy of the GNU Affero General Public License
15 // along with this program. If not, see <http://www.gnu.org/licenses/>.
16
17 use env_logger::Env;
18 use ev3_rust::{Peripherals, RobotState};
19 use ev3dev_lang_rust::{
20     motors::LargeMotor,
21     sensors::{GyroSensor, UltrasonicSensor},
22     Ev3Error, Led,
23 };
24 use log::info;
25
26 fn main() -> Result<(), Ev3Error> {
27     env_logger::init_from_env(Env::new().default_filter_or("trace"));
28
29     info!("hello!");
30
31     let mut peripherals = Peripherals {
32         drive: LargeMotor::find()
33             .expect("failed to find main drive motor, or there is more than one"),
34         gyroscope: GyroSensor::find().expect("failed to find gyroscope, or there is
more than one"),
35         ultrasonic: UltrasonicSensor::find()
36             .expect("failed to find ultrasonic sensor, or there is more than one"),
37         led: Led::new()?,
38     };
39
40     let mut robot = RobotState::default();
41
42     robot.setup(&mut peripherals)?;
43     peripherals.calibrate_gyroscope()?;
44     robot.forwards(&mut peripherals)?;
45
46     loop {
47         robot.update_sensor_data(&peripherals)?;
48
49         while robot.sensor_data.angle >= 35 {
50             robot.stop(&mut peripherals)?;
51         }
52
53         while robot.sensor_data.distance <= 10.0 {
54             robot.stop(&mut peripherals)?;
55         }
56     }

```

```

57         robot.forwards(&mut peripherals)?;
58     }
59 }
60

```

rust

ev3-rust/src/lib.rs

```

1 // ev3-rust
2 // Copyright (C) 2025 skoove
3
4 // This program is free software: you can redistribute it and/or modify
5 // it under the terms of the GNU Affero General Public License as published by
6 // the Free Software Foundation, either version 3 of the License, or
7 // (at your option) any later version.
8
9 // This program is distributed in the hope that it will be useful,
10 // but WITHOUT ANY WARRANTY; without even the implied warranty of
11 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 // GNU Affero General Public License for more details.
13
14 // You should have received a copy of the GNU Affero General Public License
15 // along with this program. If not, see <http://www.gnu.org/licenses/>.
16
17 use ev3dev_lang_rust::{motors, sensors, Ev3Error, Led};
18 use log::info;
19 use std::{thread::sleep, time::Duration};
20
21 #[derive(Default, PartialEq, Eq, Clone, Copy, Debug)]
22 pub enum MoveState {
23     #[default]
24     Stop,
25     Forwards,
26     Backwards,
27 }
28
29 #[derive(Default, Clone, Copy, Debug)]
30 pub struct RobotState {
31     move_state: MoveState,
32     pub sensor_data: SensorData,
33 }
34
35 #[derive(Default, Clone, Copy, Debug)]
36 pub struct SensorData {
37     pub angle: i32,
38     pub distance: f32,
39 }
40
41 pub struct Peripherals {
42     pub drive: motors::LargeMotor,
43     pub gyroscope: sensors::GyroSensor,
44     pub ultrasonic: sensors::UltrasonicSensor,

```

```

45     pub led: Led,
46 }
47
48 impl Peripherals {
49     /// Calibartes gyroscope
50     /// Sets led to red, sets the gyro to cal mode then sleeps for a second,
51     /// then turns LED green again and reutrns Ok(())
52     pub fn calibrate_gyroscope(&mut self) -> Result<(), Ev3Error> {
53         self.led.set_color(Led::COLOR_RED)?;
54         info!("gyroscope calibration begin");
55
56         self.gyroscope.set_mode_gyro_cal()?;
57         sleep(Duration::from_secs(1));
58         self.gyroscope.set_mode_gyro_ang()?;
59
60         self.led.set_color(Led::COLOR_GREEN)?;
61         info!("gyroscope calibration done");
62         Ok(())
63     }
64 }
65
66 impl RobotState {
67     pub fn stop(
68         &mut self,
69         peripherals: &mut Peripherals,
70     ) -> Result<(), ev3dev_lang_rust::Ev3Error> {
71         if self.move_state == MoveState::Stop {
72             return Ok(());
73         }
74
75         info!("stop");
76         self.move_state = MoveState::Stop;
77         peripherals.drive.set_duty_cycle_sp(0)
78     }
79
80     pub fn forwards(
81         &mut self,
82         peripherals: &mut Peripherals,
83     ) -> Result<(), ev3dev_lang_rust::Ev3Error> {
84         if self.move_state == MoveState::Forwards {
85             return Ok(());
86         }
87
88         info!("moving forwards");
89         if self.move_state == MoveState::Stop {
90             self.move_state = MoveState::Forwards;
91             peripherals.drive.set_duty_cycle_sp(100)?;
92         } else {
93             info!("tried to go forwards while moving, stopping first");
94             sleep(Duration::from_millis(500));

```

```

95         self.stop(peripherals)?;
96         self.move_state = MoveState::Forwards;
97         peripherals.drive.set_duty_cycle_sp(-100)?;
98     }
99
100    Ok(())
101 }
102
103    pub fn backwards(
104        &mut self,
105        peripherals: &mut Peripherals,
106    ) -> Result<(), ev3dev_lang_rust::Ev3Error> {
107        info!("moving backwards");
108        if self.move_state == MoveState::Backwards {
109            return Ok(());
110        }
111
112        if self.move_state == MoveState::Stop {
113            self.move_state = MoveState::Backwards;
114            peripherals.drive.set_duty_cycle_sp(100)?;
115        } else {
116            info!("tried to go forwards while moving, stopping first");
117            sleep(Duration::from_millis(500));
118            self.stop(peripherals)?;
119            self.move_state = MoveState::Backwards;
120            peripherals.drive.set_duty_cycle_sp(-100)?;
121        }
122
123    Ok(())
124 }
125
126    pub fn update_sensor_data(&mut self, peripherals: &Peripherals) -> Result<(), Ev3Error> {
127        self.sensor_data.angle = peripherals.gyroscope.get_angle()?;
128        self.sensor_data.distance =
129        peripherals.ultrasonic.get_distance_centimeters()?;
130        Ok(())
131    }
132
133    pub fn setup(&mut self, peripherals: &mut Peripherals) -> Result<(), Ev3Error> {
134        peripherals.drive.run_direct()?;
135        peripherals.ultrasonic.set_mode_us_dist_cm()?;
136    }
137 }
138

```

txt

ev3-rust/LICENSE

```

1      GNU AFFERO GENERAL PUBLIC LICENSE
2      Version 3, 19 November 2007

```

3 Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>
4 Everyone is permitted to copy and distribute verbatim copies
5 of this license document, but changing it is not allowed.
6
7 Preamble
8
9 The GNU Affero General Public License is a free, copyleft license for
10 software and other kinds of works, specifically designed to ensure
11 cooperation with the community in the case of network server software.
12
13 The licenses for most software and other practical works are designed
14 to take away your freedom to share and change the works. By contrast,
15 our General Public Licenses are intended to guarantee your freedom to
16 share and change all versions of a program--to make sure it remains free
17 software for all its users.
18
19 When we speak of free software, we are referring to freedom, not
20 price. Our General Public Licenses are designed to make sure that you
21 have the freedom to distribute copies of free software (and charge for
22 them if you wish), that you receive source code or can get it if you
23 want it, that you can change the software or use pieces of it in new
24 free programs, and that you know you can do these things.
25
26 Developers that use our General Public Licenses protect your rights
27 with two steps: (1) assert copyright on the software, and (2) offer
28 you this License which gives you legal permission to copy, distribute
29 and/or modify the software.
30
31 A secondary benefit of defending all users' freedom is that
32 improvements made in alternate versions of the program, if they
33 receive widespread use, become available for other developers to
34 incorporate. Many developers of free software are heartened and
35 encouraged by the resulting cooperation. However, in the case of
36 software used on network servers, this result may fail to come about.
37 The GNU General Public License permits making a modified version and
38 letting the public access it on a server without ever releasing its
39 source code to the public.
40
41 The GNU Affero General Public License is designed specifically to
42 ensure that, in such cases, the modified source code becomes available
43 to the community. It requires the operator of a network server to
44 provide the source code of the modified version running there to the
45 users of that server. Therefore, public use of a modified version, on
46 a publicly accessible server, gives the public access to the source
47 code of the modified version.
48
49 An older license, called the Affero General Public License and
50 published by Affero, was designed to accomplish similar goals. This is
51 a different license, not a version of the Affero GPL, but Affero has

53 released a new version of the Affero GPL which permits relicensing under
54 this license.
55
56 The precise terms and conditions for copying, distribution and
57 modification follow.
58
59 TERMS AND CONDITIONS
60
61 0. Definitions.
62
63 "This License" refers to version 3 of the GNU Affero General Public License.
64
65 "Copyright" also means copyright-like laws that apply to other kinds of
66 works, such as semiconductor masks.
67
68 "The Program" refers to any copyrightable work licensed under this
69 License. Each licensee is addressed as "you". "Licensees" and
70 "recipients" may be individuals or organizations.
71
72 To "modify" a work means to copy from or adapt all or part of the work
73 in a fashion requiring copyright permission, other than the making of an
74 exact copy. The resulting work is called a "modified version" of the
75 earlier work or a work "based on" the earlier work.
76
77 A "covered work" means either the unmodified Program or a work based
78 on the Program.
79
80 To "propagate" a work means to do anything with it that, without
81 permission, would make you directly or secondarily liable for
82 infringement under applicable copyright law, except executing it on a
83 computer or modifying a private copy. Propagation includes copying,
84 distribution (with or without modification), making available to the
85 public, and in some countries other activities as well.
86
87 To "convey" a work means any kind of propagation that enables other
88 parties to make or receive copies. Mere interaction with a user through
89 a computer network, with no transfer of a copy, is not conveying.
90
91 An interactive user interface displays "Appropriate Legal Notices"
92 to the extent that it includes a convenient and prominently visible
93 feature that (1) displays an appropriate copyright notice, and (2)
94 tells the user that there is no warranty for the work (except to the
95 extent that warranties are provided), that licensees may convey the
96 work under this License, and how to view a copy of this License. If
97 the interface presents a list of user commands or options, such as a
98 menu, a prominent item in the list meets this criterion.
99
100 1. Source Code.
101
102 The "source code" for a work means the preferred form of the work

103 for making modifications to it. "Object code" means any non-source
104 form of a work.

105

106 A "Standard Interface" means an interface that either is an official
107 standard defined by a recognized standards body, or, in the case of
108 interfaces specified for a particular programming language, one that
109 is widely used among developers working in that language.

110

111 The "System Libraries" of an executable work include anything, other
112 than the work as a whole, that (a) is included in the normal form of
113 packaging a Major Component, but which is not part of that Major
114 Component, and (b) serves only to enable use of the work with that
115 Major Component, or to implement a Standard Interface for which an
116 implementation is available to the public in source code form. A
117 "Major Component", in this context, means a major essential component
118 (kernel, window system, and so on) of the specific operating system
119 (if any) on which the executable work runs, or a compiler used to
120 produce the work, or an object code interpreter used to run it.

121

122 The "Corresponding Source" for a work in object code form means all
123 the source code needed to generate, install, and (for an executable
124 work) run the object code and to modify the work, including scripts to
125 control those activities. However, it does not include the work's
126 System Libraries, or general-purpose tools or generally available free
127 programs which are used unmodified in performing those activities but
128 which are not part of the work. For example, Corresponding Source
129 includes interface definition files associated with source files for
130 the work, and the source code for shared libraries and dynamically
131 linked subprograms that the work is specifically designed to require,
132 such as by intimate data communication or control flow between those
133 subprograms and other parts of the work.

134

135 The Corresponding Source need not include anything that users
136 can regenerate automatically from other parts of the Corresponding
137 Source.

138

139 The Corresponding Source for a work in source code form is that
140 same work.

141

142 2. Basic Permissions.

143

144 All rights granted under this License are granted for the term of
145 copyright on the Program, and are irrevocable provided the stated
146 conditions are met. This License explicitly affirms your unlimited
147 permission to run the unmodified Program. The output from running a
148 covered work is covered by this License only if the output, given its
149 content, constitutes a covered work. This License acknowledges your
150 rights of fair use or other equivalent, as provided by copyright law.

151

152 You may make, run and propagate covered works that you do not

153 convey, without conditions so long as your license otherwise remains
154 in force. You may convey covered works to others for the sole purpose
155 of having them make modifications exclusively for you, or provide you
156 with facilities for running those works, provided that you comply with
157 the terms of this License in conveying all material for which you do
158 not control copyright. Those thus making or running the covered works
159 for you must do so exclusively on your behalf, under your direction
160 and control, on terms that prohibit them from making any copies of
161 your copyrighted material outside their relationship with you.

162

163 Conveying under any other circumstances is permitted solely under
164 the conditions stated below. Sublicensing is not allowed; section 10
165 makes it unnecessary.

166

167 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

168

169 No covered work shall be deemed part of an effective technological
170 measure under any applicable law fulfilling obligations under article
171 11 of the WIPO copyright treaty adopted on 20 December 1996, or
172 similar laws prohibiting or restricting circumvention of such
173 measures.

174

175 When you convey a covered work, you waive any legal power to forbid
176 circumvention of technological measures to the extent such circumvention
177 is effected by exercising rights under this License with respect to
178 the covered work, and you disclaim any intention to limit operation or
179 modification of the work as a means of enforcing, against the work's
180 users, your or third parties' legal rights to forbid circumvention of
181 technological measures.

182

183 4. Conveying Verbatim Copies.

184

185 You may convey verbatim copies of the Program's source code as you
186 receive it, in any medium, provided that you conspicuously and
187 appropriately publish on each copy an appropriate copyright notice;
188 keep intact all notices stating that this License and any
189 non-permissive terms added in accord with section 7 apply to the code;
190 keep intact all notices of the absence of any warranty; and give all
191 recipients a copy of this License along with the Program.

192

193 You may charge any price or no price for each copy that you convey,
194 and you may offer support or warranty protection for a fee.

195

196 5. Conveying Modified Source Versions.

197

198 You may convey a work based on the Program, or the modifications to
199 produce it from the Program, in the form of source code under the
200 terms of section 4, provided that you also meet all of these conditions:

201

202 a) The work must carry prominent notices stating that you modified

203 it, and giving a relevant date.
204
205 b) The work must carry prominent notices stating that it is
206 released under this License and any conditions added under section
207 7. This requirement modifies the requirement in section 4 to
208 "keep intact all notices".
209
210 c) You must license the entire work, as a whole, under this
211 License to anyone who comes into possession of a copy. This
212 License will therefore apply, along with any applicable section 7
213 additional terms, to the whole of the work, and all its parts,
214 regardless of how they are packaged. This License gives no
215 permission to license the work in any other way, but it does not
216 invalidate such permission if you have separately received it.
217
218 d) If the work has interactive user interfaces, each must display
219 Appropriate Legal Notices; however, if the Program has interactive
220 interfaces that do not display Appropriate Legal Notices, your
221 work need not make them do so.
222
223 A compilation of a covered work with other separate and independent
224 works, which are not by their nature extensions of the covered work,
225 and which are not combined with it such as to form a larger program,
226 in or on a volume of a storage or distribution medium, is called an
227 "aggregate" if the compilation and its resulting copyright are not
228 used to limit the access or legal rights of the compilation's users
229 beyond what the individual works permit. Inclusion of a covered work
230 in an aggregate does not cause this License to apply to the other
231 parts of the aggregate.
232
233 6. Conveying Non-Source Forms.
234
235 You may convey a covered work in object code form under the terms
236 of sections 4 and 5, provided that you also convey the
237 machine-readable Corresponding Source under the terms of this License,
238 in one of these ways:
239
240 a) Convey the object code in, or embodied in, a physical product
241 (including a physical distribution medium), accompanied by the
242 Corresponding Source fixed on a durable physical medium
243 customarily used for software interchange.
244
245 b) Convey the object code in, or embodied in, a physical product
246 (including a physical distribution medium), accompanied by a
247 written offer, valid for at least three years and valid for as
248 long as you offer spare parts or customer support for that product
249 model, to give anyone who possesses the object code either (1) a
250 copy of the Corresponding Source for all the software in the
251 product that is covered by this License, on a durable physical
252 medium customarily used for software interchange, for a price no

253 more than your reasonable cost of physically performing this
254 conveying of source, or (2) access to copy the
255 Corresponding Source from a network server at no charge.
256
257 c) Convey individual copies of the object code with a copy of the
258 written offer to provide the Corresponding Source. This
259 alternative is allowed only occasionally and noncommercially, and
260 only if you received the object code with such an offer, in accord
261 with subsection 6b.
262
263 d) Convey the object code by offering access from a designated
264 place (gratis or for a charge), and offer equivalent access to the
265 Corresponding Source in the same way through the same place at no
266 further charge. You need not require recipients to copy the
267 Corresponding Source along with the object code. If the place to
268 copy the object code is a network server, the Corresponding Source
269 may be on a different server (operated by you or a third party)
270 that supports equivalent copying facilities, provided you maintain
271 clear directions next to the object code saying where to find the
272 Corresponding Source. Regardless of what server hosts the
273 Corresponding Source, you remain obligated to ensure that it is
274 available for as long as needed to satisfy these requirements.
275
276 e) Convey the object code using peer-to-peer transmission, provided
277 you inform other peers where the object code and Corresponding
278 Source of the work are being offered to the general public at no
279 charge under subsection 6d.
280
281 A separable portion of the object code, whose source code is excluded
282 from the Corresponding Source as a System Library, need not be
283 included in conveying the object code work.
284
285 A "User Product" is either (1) a "consumer product", which means any
286 tangible personal property which is normally used for personal, family,
287 or household purposes, or (2) anything designed or sold for incorporation
288 into a dwelling. In determining whether a product is a consumer product,
289 doubtful cases shall be resolved in favor of coverage. For a particular
290 product received by a particular user, "normally used" refers to a
291 typical or common use of that class of product, regardless of the status
292 of the particular user or of the way in which the particular user
293 actually uses, or expects or is expected to use, the product. A product
294 is a consumer product regardless of whether the product has substantial
295 commercial, industrial or non-consumer uses, unless such uses represent
296 the only significant mode of use of the product.
297
298 "Installation Information" for a User Product means any methods,
299 procedures, authorization keys, or other information required to install
300 and execute modified versions of a covered work in that User Product from
301 a modified version of its Corresponding Source. The information must
302 suffice to ensure that the continued functioning of the modified object

303 code is in no case prevented or interfered with solely because
304 modification has been made.

305

306 If you convey an object code work under this section in, or with, or
307 specifically for use in, a User Product, and the conveying occurs as
308 part of a transaction in which the right of possession and use of the
309 User Product is transferred to the recipient in perpetuity or for a
310 fixed term (regardless of how the transaction is characterized), the
311 Corresponding Source conveyed under this section must be accompanied
312 by the Installation Information. But this requirement does not apply
313 if neither you nor any third party retains the ability to install
314 modified object code on the User Product (for example, the work has
315 been installed in ROM).

316

317 The requirement to provide Installation Information does not include a
318 requirement to continue to provide support service, warranty, or updates
319 for a work that has been modified or installed by the recipient, or for
320 the User Product in which it has been modified or installed. Access to a
321 network may be denied when the modification itself materially and
322 adversely affects the operation of the network or violates the rules and
323 protocols for communication across the network.

324

325 Corresponding Source conveyed, and Installation Information provided,
326 in accord with this section must be in a format that is publicly
327 documented (and with an implementation available to the public in
328 source code form), and must require no special password or key for
329 unpacking, reading or copying.

330

331 7. Additional Terms.

332

333 "Additional permissions" are terms that supplement the terms of this
334 License by making exceptions from one or more of its conditions.
335 Additional permissions that are applicable to the entire Program shall
336 be treated as though they were included in this License, to the extent
337 that they are valid under applicable law. If additional permissions
338 apply only to part of the Program, that part may be used separately
339 under those permissions, but the entire Program remains governed by
340 this License without regard to the additional permissions.

341

342 When you convey a copy of a covered work, you may at your option
343 remove any additional permissions from that copy, or from any part of
344 it. (Additional permissions may be written to require their own
345 removal in certain cases when you modify the work.) You may place
346 additional permissions on material, added by you to a covered work,
347 for which you have or can give appropriate copyright permission.

348

349 Notwithstanding any other provision of this License, for material you
350 add to a covered work, you may (if authorized by the copyright holders of
351 that material) supplement the terms of this License with terms:

352

353 a) Disclaiming warranty or limiting liability differently from the
354 terms of sections 15 and 16 of this License; or

355

356 b) Requiring preservation of specified reasonable legal notices or
357 author attributions in that material or in the Appropriate Legal
358 Notices displayed by works containing it; or

359

360 c) Prohibiting misrepresentation of the origin of that material, or
361 requiring that modified versions of such material be marked in
362 reasonable ways as different from the original version; or

363

364 d) Limiting the use for publicity purposes of names of licensors or
365 authors of the material; or

366

367 e) Declining to grant rights under trademark law for use of some
368 trade names, trademarks, or service marks; or

369

370 f) Requiring indemnification of licensors and authors of that
371 material by anyone who conveys the material (or modified versions of
372 it) with contractual assumptions of liability to the recipient, for
373 any liability that these contractual assumptions directly impose on
374 those licensors and authors.

375

376 All other non-permissive additional terms are considered "further
377 restrictions" within the meaning of section 10. If the Program as you
378 received it, or any part of it, contains a notice stating that it is
379 governed by this License along with a term that is a further
380 restriction, you may remove that term. If a license document contains
381 a further restriction but permits relicensing or conveying under this
382 License, you may add to a covered work material governed by the terms
383 of that license document, provided that the further restriction does
384 not survive such relicensing or conveying.

385

386 If you add terms to a covered work in accord with this section, you
387 must place, in the relevant source files, a statement of the
388 additional terms that apply to those files, or a notice indicating
389 where to find the applicable terms.

390

391 Additional terms, permissive or non-permissive, may be stated in the
392 form of a separately written license, or stated as exceptions;
393 the above requirements apply either way.

394

395 8. Termination.

396

397 You may not propagate or modify a covered work except as expressly
398 provided under this License. Any attempt otherwise to propagate or
399 modify it is void, and will automatically terminate your rights under
400 this License (including any patent licenses granted under the third
401 paragraph of section 11).

402

403 However, if you cease all violation of this License, then your
404 license from a particular copyright holder is reinstated (a)
405 provisionally, unless and until the copyright holder explicitly and
406 finally terminates your license, and (b) permanently, if the copyright
407 holder fails to notify you of the violation by some reasonable means
408 prior to 60 days after the cessation.

409
410 Moreover, your license from a particular copyright holder is
411 reinstated permanently if the copyright holder notifies you of the
412 violation by some reasonable means, this is the first time you have
413 received notice of violation of this License (for any work) from that
414 copyright holder, and you cure the violation prior to 30 days after
415 your receipt of the notice.

416
417 Termination of your rights under this section does not terminate the
418 licenses of parties who have received copies or rights from you under
419 this License. If your rights have been terminated and not permanently
420 reinstated, you do not qualify to receive new licenses for the same
421 material under section 10.

422
423 9. Acceptance Not Required for Having Copies.
424

425 You are not required to accept this License in order to receive or
426 run a copy of the Program. Ancillary propagation of a covered work
427 occurring solely as a consequence of using peer-to-peer transmission
428 to receive a copy likewise does not require acceptance. However,
429 nothing other than this License grants you permission to propagate or
430 modify any covered work. These actions infringe copyright if you do
431 not accept this License. Therefore, by modifying or propagating a
432 covered work, you indicate your acceptance of this License to do so.

433
434 10. Automatic Licensing of Downstream Recipients.
435

436 Each time you convey a covered work, the recipient automatically
437 receives a license from the original licensors, to run, modify and
438 propagate that work, subject to this License. You are not responsible
439 for enforcing compliance by third parties with this License.

440
441 An "entity transaction" is a transaction transferring control of an
442 organization, or substantially all assets of one, or subdividing an
443 organization, or merging organizations. If propagation of a covered
444 work results from an entity transaction, each party to that
445 transaction who receives a copy of the work also receives whatever
446 licenses to the work the party's predecessor in interest had or could
447 give under the previous paragraph, plus a right to possession of the
448 Corresponding Source of the work from the predecessor in interest, if
449 the predecessor has it or can get it with reasonable efforts.

450
451 You may not impose any further restrictions on the exercise of the
452 rights granted or affirmed under this License. For example, you may

453 not impose a license fee, royalty, or other charge for exercise of
454 rights granted under this License, and you may not initiate litigation
455 (including a cross-claim or counterclaim in a lawsuit) alleging that
456 any patent claim is infringed by making, using, selling, offering for
457 sale, or importing the Program or any portion of it.

458
459 11. Patents.
460

461 A "contributor" is a copyright holder who authorizes use under this
462 License of the Program or a work on which the Program is based. The
463 work thus licensed is called the contributor's "contributor version".

464
465 A contributor's "essential patent claims" are all patent claims
466 owned or controlled by the contributor, whether already acquired or
467 hereafter acquired, that would be infringed by some manner, permitted
468 by this License, of making, using, or selling its contributor version,
469 but do not include claims that would be infringed only as a
470 consequence of further modification of the contributor version. For
471 purposes of this definition, "control" includes the right to grant
472 patent sublicenses in a manner consistent with the requirements of
473 this License.

474
475 Each contributor grants you a non-exclusive, worldwide, royalty-free
476 patent license under the contributor's essential patent claims, to
477 make, use, sell, offer for sale, import and otherwise run, modify and
478 propagate the contents of its contributor version.

479
480 In the following three paragraphs, a "patent license" is any express
481 agreement or commitment, however denominated, not to enforce a patent
482 (such as an express permission to practice a patent or covenant not to
483 sue for patent infringement). To "grant" such a patent license to a
484 party means to make such an agreement or commitment not to enforce a
485 patent against the party.

486
487 If you convey a covered work, knowingly relying on a patent license,
488 and the Corresponding Source of the work is not available for anyone
489 to copy, free of charge and under the terms of this License, through a
490 publicly available network server or other readily accessible means,
491 then you must either (1) cause the Corresponding Source to be so
492 available, or (2) arrange to deprive yourself of the benefit of the
493 patent license for this particular work, or (3) arrange, in a manner
494 consistent with the requirements of this License, to extend the patent
495 license to downstream recipients. "Knowingly relying" means you have
496 actual knowledge that, but for the patent license, your conveying the
497 covered work in a country, or your recipient's use of the covered work
498 in a country, would infringe one or more identifiable patents in that
499 country that you have reason to believe are valid.

500
501 If, pursuant to or in connection with a single transaction or
502 arrangement, you convey, or propagate by procuring conveyance of, a

503 covered work, and grant a patent license to some of the parties
504 receiving the covered work authorizing them to use, propagate, modify
505 or convey a specific copy of the covered work, then the patent license
506 you grant is automatically extended to all recipients of the covered
507 work and works based on it.

508
509 A patent license is "discriminatory" if it does not include within
510 the scope of its coverage, prohibits the exercise of, or is
511 conditioned on the non-exercise of one or more of the rights that are
512 specifically granted under this License. You may not convey a covered
513 work if you are a party to an arrangement with a third party that is
514 in the business of distributing software, under which you make payment
515 to the third party based on the extent of your activity of conveying
516 the work, and under which the third party grants, to any of the
517 parties who would receive the covered work from you, a discriminatory
518 patent license (a) in connection with copies of the covered work
519 conveyed by you (or copies made from those copies), or (b) primarily
520 for and in connection with specific products or compilations that
521 contain the covered work, unless you entered into that arrangement,
522 or that patent license was granted, prior to 28 March 2007.

523
524 Nothing in this License shall be construed as excluding or limiting
525 any implied license or other defenses to infringement that may
526 otherwise be available to you under applicable patent law.

527
528 12. No Surrender of Others' Freedom.
529

530 If conditions are imposed on you (whether by court order, agreement or
531 otherwise) that contradict the conditions of this License, they do not
532 excuse you from the conditions of this License. If you cannot convey a
533 covered work so as to satisfy simultaneously your obligations under this
534 License and any other pertinent obligations, then as a consequence you may
535 not convey it at all. For example, if you agree to terms that obligate you
536 to collect a royalty for further conveying from those to whom you convey
537 the Program, the only way you could satisfy both those terms and this
538 License would be to refrain entirely from conveying the Program.

539
540 13. Remote Network Interaction; Use with the GNU General Public License.
541

542 Notwithstanding any other provision of this License, if you modify the
543 Program, your modified version must prominently offer all users
544 interacting with it remotely through a computer network (if your version
545 supports such interaction) an opportunity to receive the Corresponding
546 Source of your version by providing access to the Corresponding Source
547 from a network server at no charge, through some standard or customary
548 means of facilitating copying of software. This Corresponding Source
549 shall include the Corresponding Source for any work covered by version 3
550 of the GNU General Public License that is incorporated pursuant to the
551 following paragraph.

552

553 Notwithstanding any other provision of this License, you have
554 permission to link or combine any covered work with a work licensed
555 under version 3 of the GNU General Public License into a single
556 combined work, and to convey the resulting work. The terms of this
557 License will continue to apply to the part which is the covered work,
558 but the work with which it is combined will remain governed by version
559 3 of the GNU General Public License.

560
561 14. Revised Versions of this License.
562

563 The Free Software Foundation may publish revised and/or new versions of
564 the GNU Affero General Public License from time to time. Such new versions
565 will be similar in spirit to the present version, but may differ in detail to
566 address new problems or concerns.

567
568 Each version is given a distinguishing version number. If the
569 Program specifies that a certain numbered version of the GNU Affero General
570 Public License "or any later version" applies to it, you have the
571 option of following the terms and conditions either of that numbered
572 version or of any later version published by the Free Software
573 Foundation. If the Program does not specify a version number of the
574 GNU Affero General Public License, you may choose any version ever published
575 by the Free Software Foundation.

576
577 If the Program specifies that a proxy can decide which future
578 versions of the GNU Affero General Public License can be used, that proxy's
579 public statement of acceptance of a version permanently authorizes you
580 to choose that version for the Program.

581
582 Later license versions may give you additional or different
583 permissions. However, no additional obligations are imposed on any
584 author or copyright holder as a result of your choosing to follow a
585 later version.

586
587 15. Disclaimer of Warranty.
588

589 THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
590 APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
591 HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
592 OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
593 THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
594 PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
595 IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
596 ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

597
598 16. Limitation of Liability.
599

600 IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
601 WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
602 THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY

603 GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
604 USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
605 DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
606 PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
607 EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
608 SUCH DAMAGES.

609
610 17. Interpretation of Sections 15 and 16.

611
612 If the disclaimer of warranty and limitation of liability provided
613 above cannot be given local legal effect according to their terms,
614 reviewing courts shall apply local law that most closely approximates
615 an absolute waiver of all civil liability in connection with the
616 Program, unless a warranty or assumption of liability accompanies a
617 copy of the Program in return for a fee.

618
619 END OF TERMS AND CONDITIONS

620
621 How to Apply These Terms to Your New Programs

622
623 If you develop a new program, and you want it to be of the greatest
624 possible use to the public, the best way to achieve this is to make it
625 free software which everyone can redistribute and change under these terms.

626
627 To do so, attach the following notices to the program. It is safest
628 to attach them to the start of each source file to most effectively
629 state the exclusion of warranty; and each file should have at least
630 the "copyright" line and a pointer to where the full notice is found.

631
632 <one line to give the program's name and a brief idea of what it does.>
633 Copyright (C) <year> <name of author>

634
635 This program is free software: you can redistribute it and/or modify
636 it under the terms of the GNU Affero General Public License as published
637 by the Free Software Foundation, either version 3 of the License, or
638 (at your option) any later version.

639
640 This program is distributed in the hope that it will be useful,
641 but WITHOUT ANY WARRANTY; without even the implied warranty of
642 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
643 GNU Affero General Public License for more details.

644
645 You should have received a copy of the GNU Affero General Public License
646 along with this program. If not, see <<https://www.gnu.org/licenses/>>.

647
648 Also add information on how to contact you by electronic and paper mail.

649
650 If your software can interact with users remotely through a computer
651 network, you should also make sure that it provides a way for users to
652 get its source. For example, if your program is a web application, its

653 interface could display a "Source" link that leads users to an archive
654 of the code. There are many ways you could offer source, and different
655 solutions will be better for different programs; see section 13 for the
656 specific requirements.

657
658 You should also get your employer (if you work as a programmer) or school,
659 if any, to sign a "copyright disclaimer" for the program, if necessary.
660 For more information on this, and how to apply and follow the GNU AGPL, see
661 <<https://www.gnu.org/licenses/>>.

662