

Making the Mars Rover Demo

Fabien Chouteau

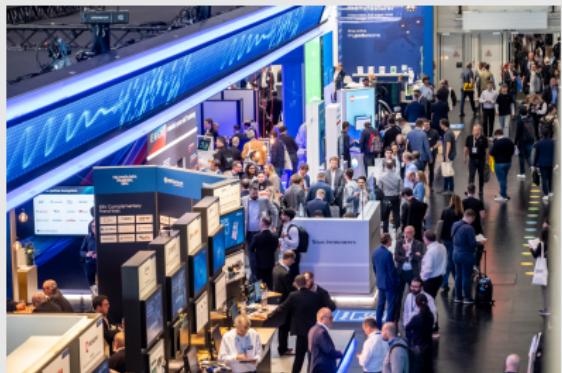
Embedded Software Engineer at AdaCore

 Mastodon : @DesChips@mamot.fr

 GitHub : Fabien-Chouteau

 Hackaday.io: Fabien.C

Context



Why do we make demos for trade shows?

Why do we make demos for trade shows?

- Software is not easy to show
-
-

Why do we make demos for trade shows?

- Software is not easy to show
- Especially software dev tools
-

Why do we make demos for trade shows?

- Software is not easy to show
- Especially software dev tools
- We have to quickly grab attention

Constraints?

- Eye catching
-
-
-

Constraints?

- Eye catching
- Easy to deploy/use
-
-

Constraints?

- Eye catching
- Easy to deploy/use
- As much as possible relevant
-

Constraints?

- Eye catching
- Easy to deploy/use
- As much as possible relevant
- Reliable...

The Process

Finding the good idea

- First idea, a remote controlled thingy
-
-

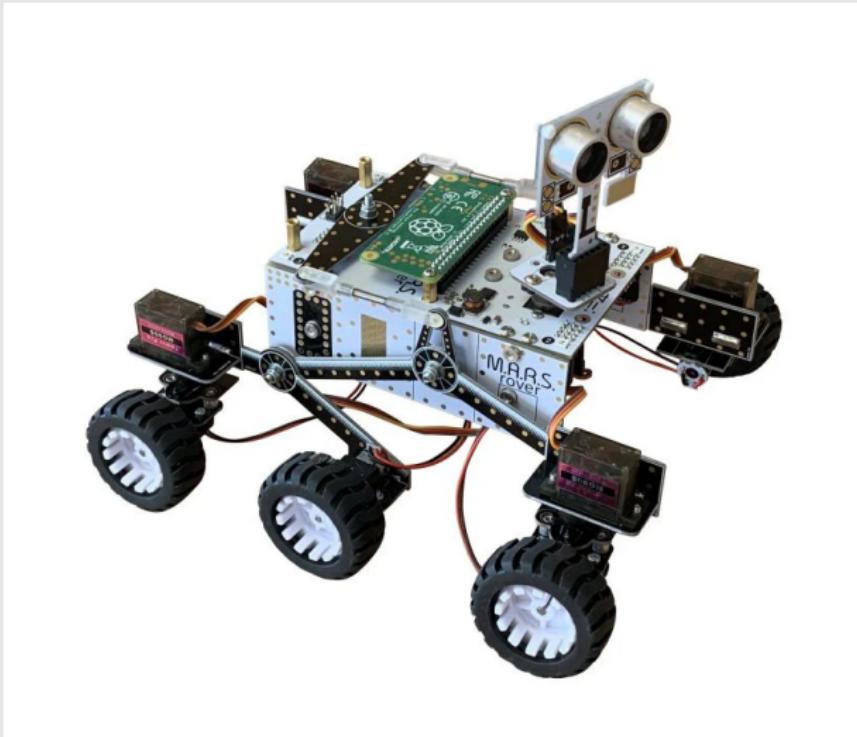
Finding the good idea

- First idea, a remote controlled thingy
- Looking for robots that would look nice
-

Finding the good idea

- First idea, a remote controlled thingy
- Looking for robots that would look nice
- Looking for robots that would be relevant

4Tronix M.A.R.S Rover



<https://shop.4tronix.co.uk/products/marsrover>

Investigate feasibility

- How to control?
-
-

Investigate feasibility

- How to control? → RPi Pico
-
-

Investigate feasibility

- How to control? → RPi Pico
- How to add remote?
-

Investigate feasibility

- How to control? → RPi Pico
- How to add remote? → Wireless PS2 controller
-

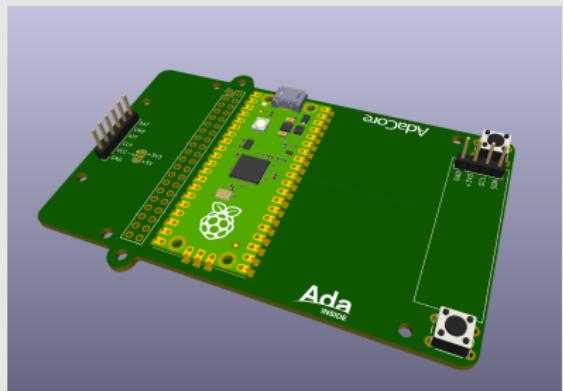
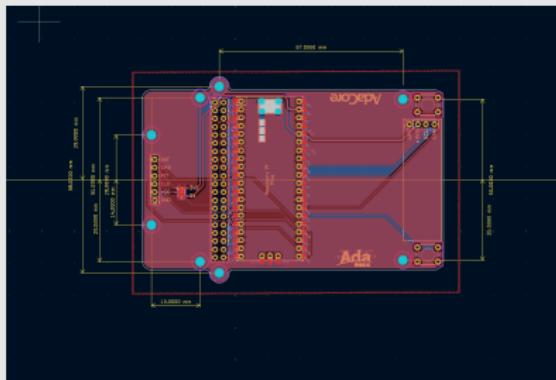
Investigate feasibility

- How to control? → RPi Pico
- How to add remote? → Wireless PS2 controller
- Autonomous mode?

Investigate feasibility

- How to control? → RPi Pico
- How to add remote? → Wireless PS2 controller
- Autonomous mode? → Onboard sensor

Implementation: A Custom PCB



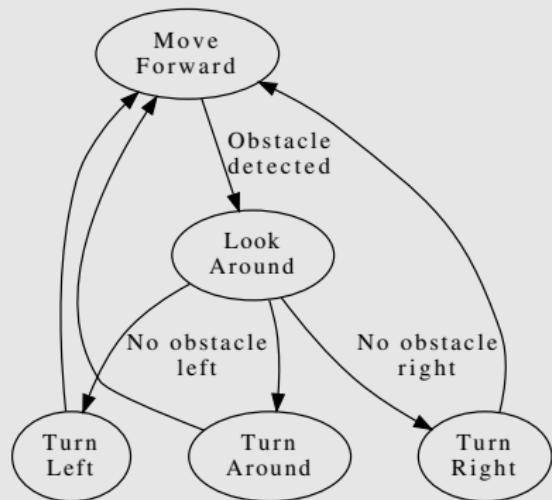
- RPi Pico
- PS2 remote
- Screen and buttons

Implementation: Software

- Using Alire of course
- rp2040-hal
- New drivers needed (Servos, and PS2 remote)

Implementation: Autonomous Mode

- Very simple (KISS)
- Never hit a wall
- Never get stuck



Implementation:

```
function Cannot_Crash return Boolean is
  (if Rover_HAL.Get_Sonar_Distance < Safety_Distance and then
    Rover_HAL.Get_Turn = Rover_HAL.Straight
  then
    Rover_HAL.Get_Power (Rover_HAL.Left)  <= 0 and then
    Rover_HAL.Get_Power (Rover_HAL.Right) <= 0);
```

```
[...]  
  
    Delay_Milliseconds (30);  
  
    pragma Loop_Invariant (Rover.Cannot_Crash);  
end loop;  
end Run;
```

```
Dist := Sonar_Distance;

if Dist < Rover.Safety_Distance then
    -- Ignore forward commands when close to an obstacle
    Buttons (Up) := False;
end if;
```

blog.adacore.com/lets-write-a-safety-monitor-for-a-mars-rover

Deploy



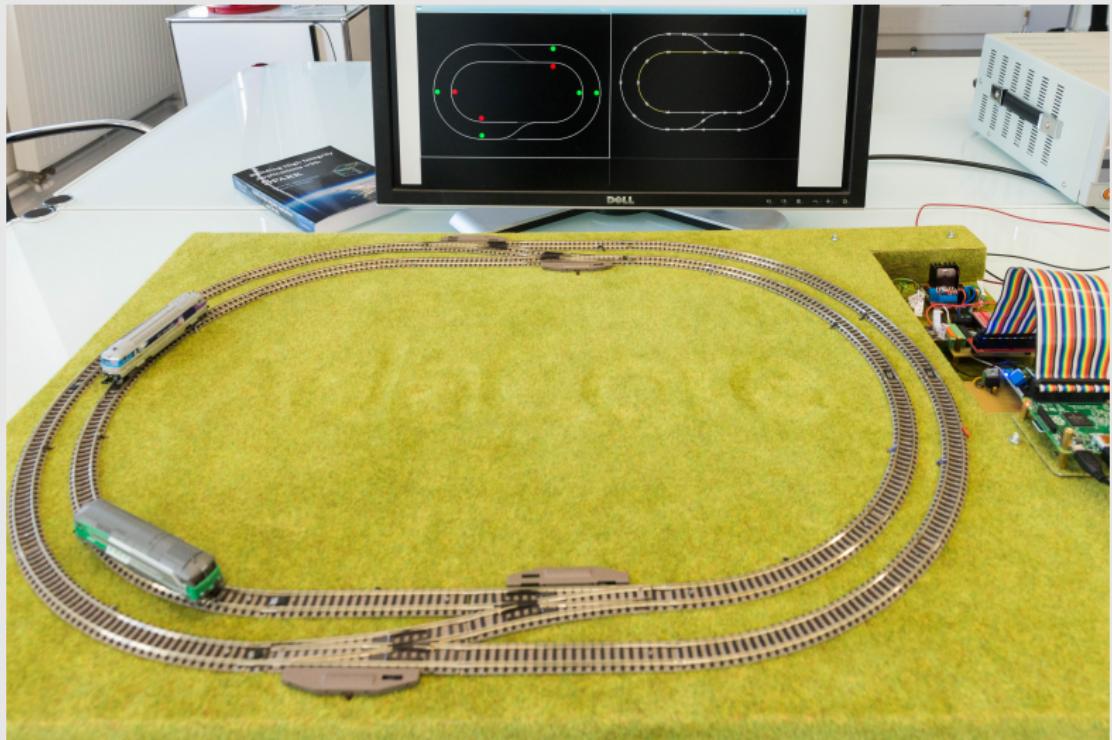
AdaCore

Add a little flag!



Other Demos

SPARK Railway



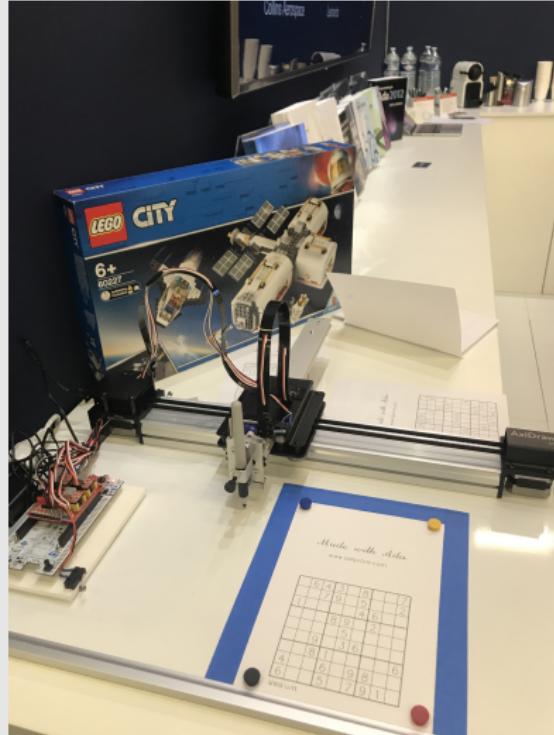
RISC-V Drawing



Crazyflie



Drawing Sudoku



Misc demos

