University of Texas at Arlington

Final Project: Stair Climbing

Project Writeup

Benjamin Knight

Rogelio Chapa

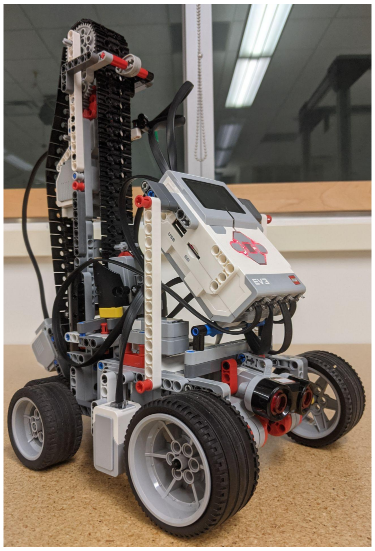
December 17, 2021

This Assignment is submitted towards and in support of the partial completion of the requirements for the Autonomous Robotics Course.

Table of Contents

Hardware Design Choices

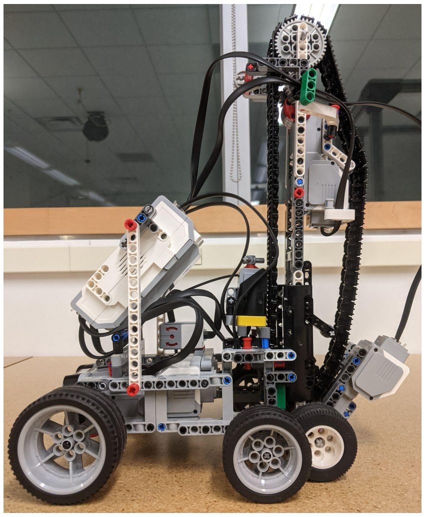
Overall Design



For our stair climbing robot we decided to use an “elevator” to lift the robot. We decided to go with this type of stair climbing robot due to our limited lego pieces and plentiful tutorials. There were some other possible robots using a scissor lift or a “rat trap” that would spring a stiff support when needed. These other robots were either too unreliable or required parts the kits we are provided with do not have.

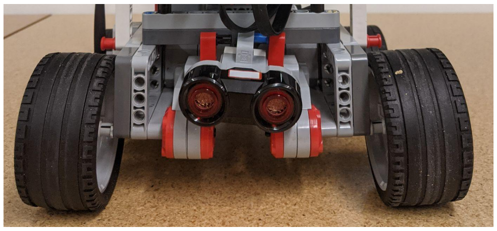
Elevator

The elevator is told to go up and down based on the gyroscope that is directly behind the EV3 brick. Once the robot has started riding up the wall it will deploy the elevator once a certain angle is reached. Because we know the length of the stairs, this ascent and descent of the elevator is timed. There is a touch sensor that is along the elevator that is used to prevent it from extending too much.



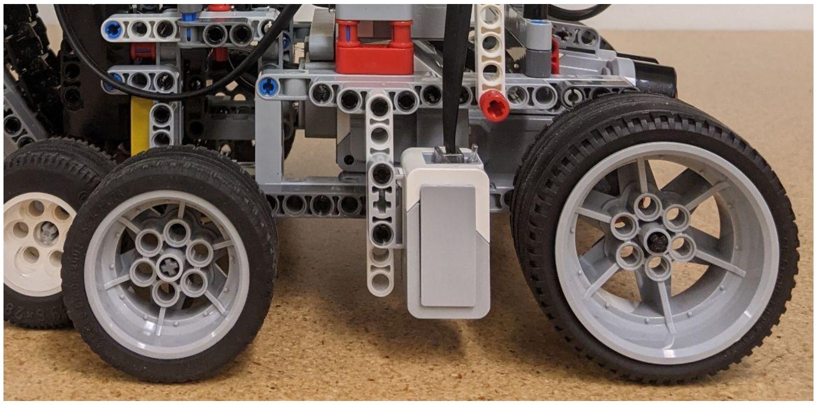
Detecting Stairs and No Stairs

To detect when we have reached the end of the stairs and avoid falling off the edge, we are using an ultrasonic sensor to detect if there is anything in front of it. This check happens after every elevator ascent/descent. This is the perfect time to check because we know that it will be level after this motion. To ensure that a true reading is found, we actively read the sensor for half a second before continuing the ascent to the stairs.



Going Down

Once going down we deploy the elevator again and start reversing. We find that this works well enough to get down the stairs. After about 8 seconds the robot will attempt to level itself and read the reflection from the color sensor mounted on the right side of the robot. We chose to use reflection instead of color because of the mottled color in the lab floor.



Software Design Choices

Why Python?

Python was a no brainer to us after having **major** problems with Eclipse not only on Windows but the Virtual Machine/Linux. It increased our productivity exponentially allowing us to code and test our algorithms rather than trying to get the development environment up and running. We did not use any libraries other than the ones required to move our robot. Theoretically you could copy our program's logic almost line by line into C and it would still work.

Algorithm

Our code has 4 main stages:

1: **Initialization,** where the robot will level itself out using this method:

```
def zero_elevator():
  elevator.run(-250)
  while(not is_pressed()):
    continue
  elevator.stop()
  elevator.run_time(250, 9000, Stop.HOLD, True)
  gyro_sens.reset_angle(0)
```

2: **Going up stairs,** the robot will move forward deploying/retracting the elevator as needed based off the gyro *until* there are no more stairs to climb

```
while(True):
  if(read_gyro() > 7):
    ev3.speaker.beep()
    elevator_up()
    elevator_down()
    if check_if_end():
      break

  move_forward()
```

3: **Going down stairs,** the robot deploys the elevator and reverses until the tiles are found

```
ev3.speaker.say("Going down stairs")
elevator_time = 3250
elevator_speed = -350
time = 0
move_elevator(elevator_speed, elevator_time)
move_forward(-100)
```

```
while(True):
  if(time > 8000):
    stop_all_motors()
    zero_elevator()
    if on_tile():
      break
    move_forward(-100)
    time = 0

  time += 5
  wait(5)
```

4: **Detecting tile,** after 8 seconds in the while loop above we will level out the robot and check to see if we are on a tile

```
def on_tile():
  return color_sens.reflection() > 25
```

Demonstration

If you'd like to see a demonstration of our project please watch this video on youtube. If you'd like a more detailed look at our code, here is the github repository.