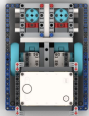


Robot Relative vs Absolute Headings

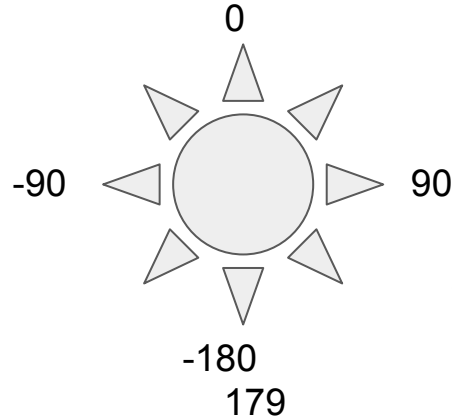
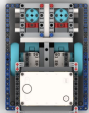
Assume the Gyro was reset when the Robot was pointing due north

These are all ABSOLUTE headings

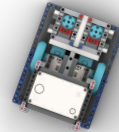
This robot is on a heading of 0 degrees.



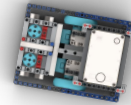
Reset here



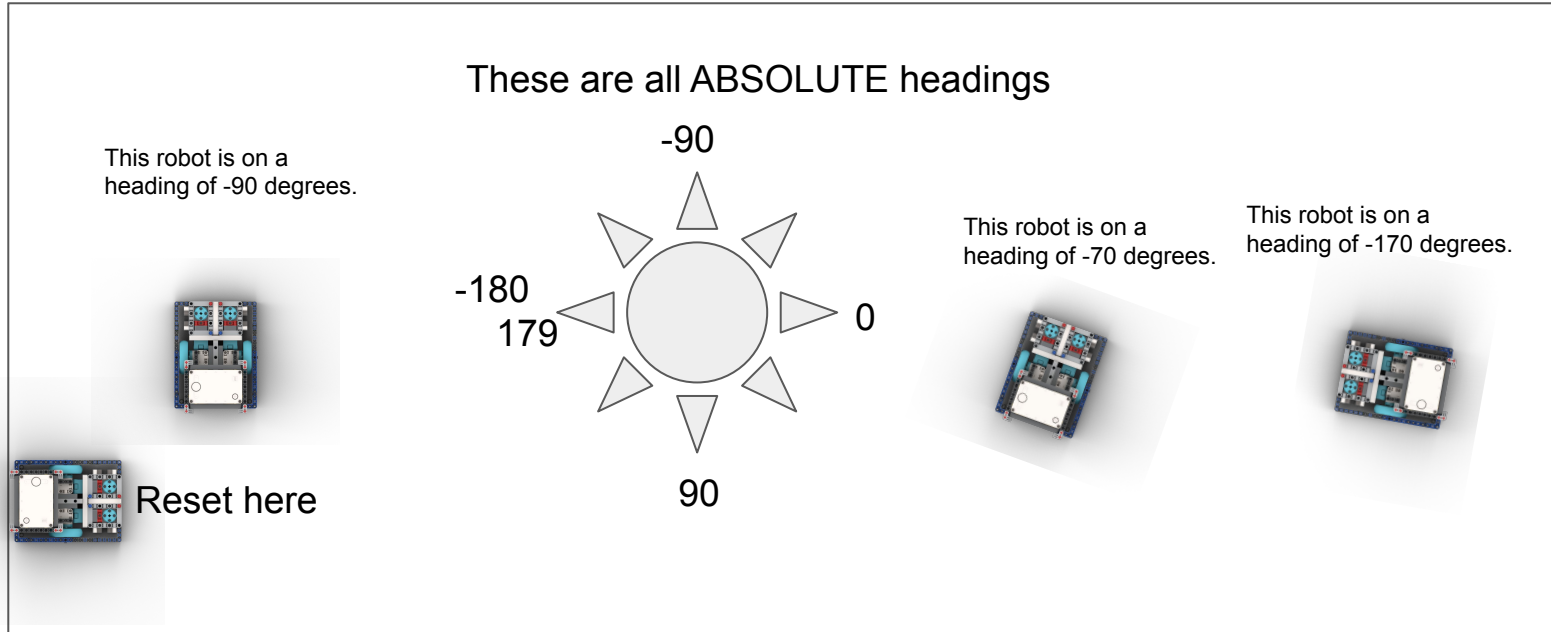
This robot is on a heading of 20 degrees.



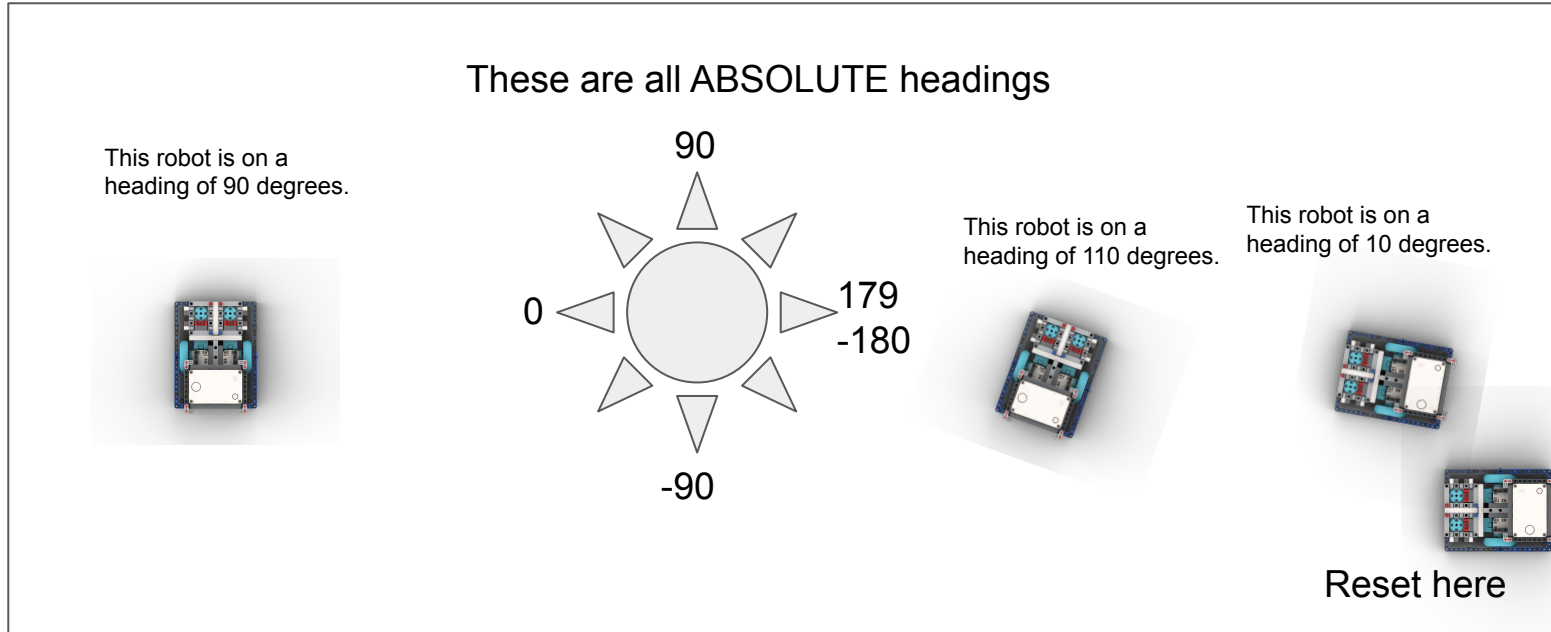
This robot is on a heading of -80 degrees.



Assume the Gyro was reset when the Robot was pointing due east



Assume the Gyro was reset when the Robot was pointing due west



What if the Gyro was reset when the Robot was pointing due south?

These are all ABSOLUTE headings

179
-180

90

-90

0

This robot is on a heading of -180 degrees (a bit of a trick question, which will be explained later)

Reset here

This robot is on a heading of -160 degrees.

This robot is on a heading of 100 degrees.

The diagram illustrates absolute headings on a circular scale. A central circle is surrounded by eight triangles pointing outwards. The headings are labeled as follows: 179 at the top-left, -180 at the top, 90 at the left, -90 at the right, and 0 at the bottom. Four robot images are shown at different orientations: one at -180 degrees (labeled as a trick question), one at -160 degrees, one at 100 degrees, and one at the top labeled 'Reset here'.

Question:

Does it matter which way the robot turned when reading the absolute heading?

Absolute headings are “attached to the table”

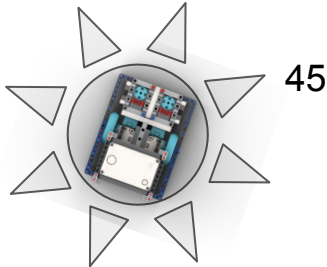
The absolute headings are reset whenever we call “reset_yaw_angle()”. The only time we should ever reset the gyro is when we are wall squared or in a jig.

If you reset the yaw angle right after a turn, you don’t know that the robot is pointing exactly where you think it is.

For example, if you reset the gyro when you *think* the robot is pointed at 20 degrees, but it is actually pointed at 23 degrees, all of your future angles will be off by 3 degrees (until you reset the gyro again, presumably when the robot is wall squared)

What’s more, the gyro ONLY sees absolute headings. If you turn the robot, it will return the heading that the gyro sees, which is going to be based on the heading the robot was pointed on when the gyro was last reset.

Relative “headings”



There are no relative headings.
There are relative TURNS.

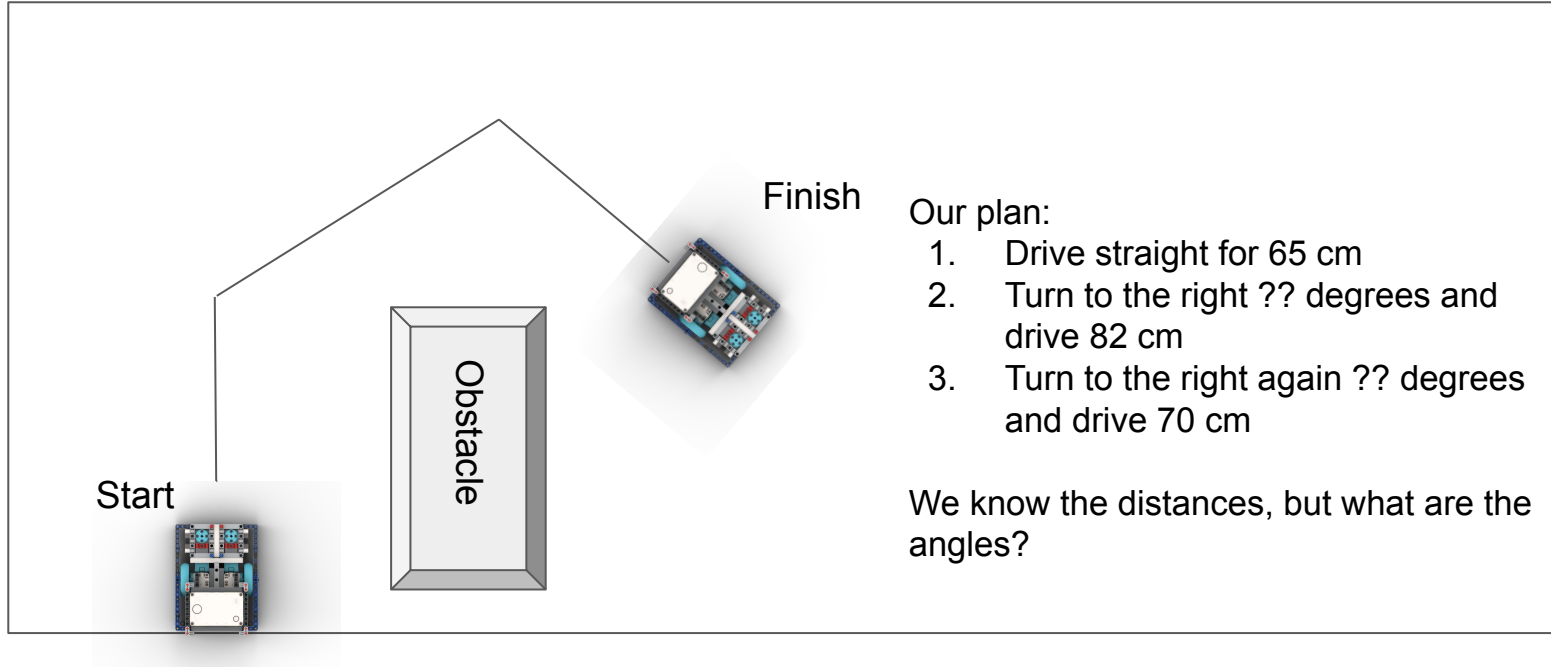
Let's say this robot reset the gyro when it was pointed due north. It has now turned to heading 20 (absolute). If you wanted it to turn “45 degrees to the right”, that is 45 RELATIVE degrees because it is RELATIVE to the current heading. The new heading will be $20 + 45 = 75$ absolute.

Are absolute headings better?

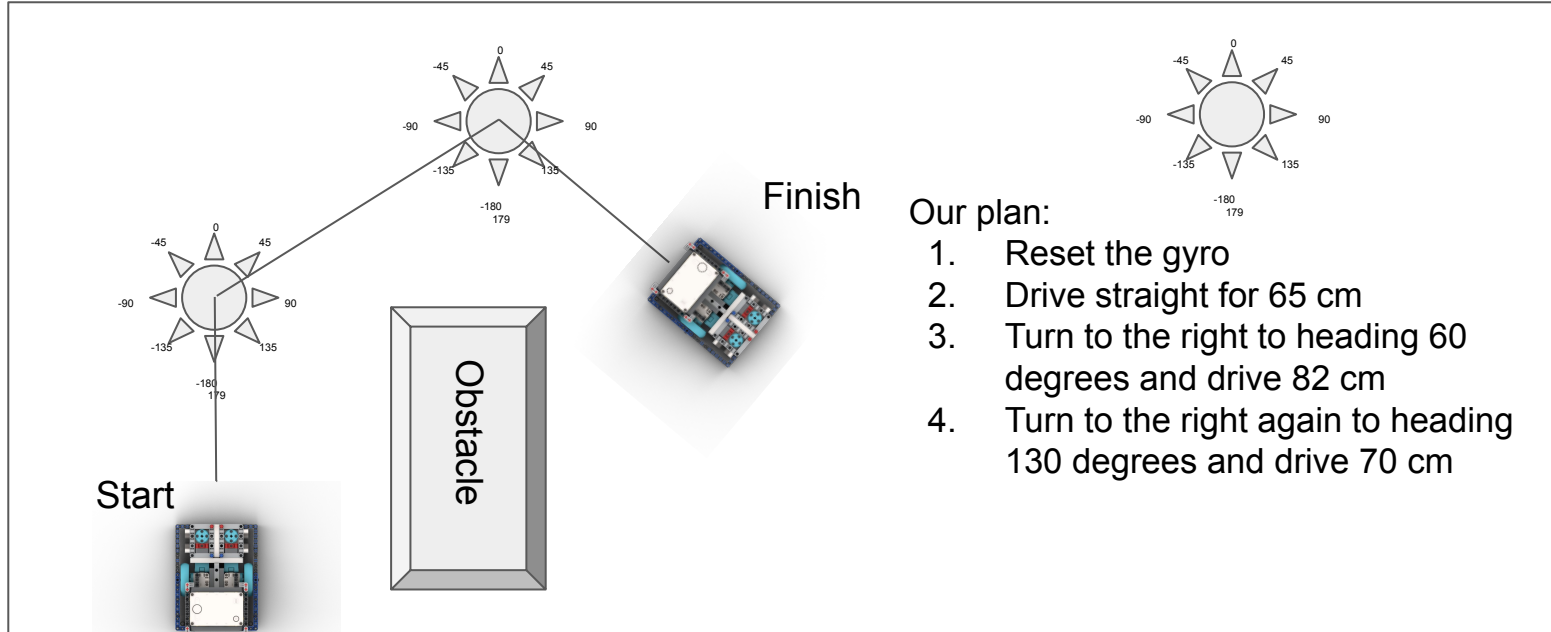
Yes and no. Imagine that you sent this robot out and commanded a relative 20 degree turn. But the robot actually only turned 16 degrees when it stopped. You asked for 20, but you got 16. If you then commanded another relative turn (say 45 degrees), well since you started at 16, you have that inaccuracy already working against you. Wouldn't it be better if you just told the robot to go to heading 75? That way it wouldn't matter where you start.

Relative headings are better because they are easier to think of, sometimes. I know what a 45 degree turn looks like, or a 90.

Sample problem



We know the angles!



What would the code look like?

```
br = BaseRobot()
```

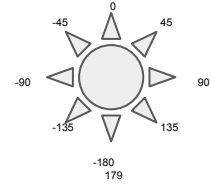
```
br.hub.motion_sensor.reset_yaw_angle()
```

```
br.AccelGyroDrive(60) # drive straight for 60 cm
```

```
br.AbsTurnRightAndDriveOnHeading(heading=60, distance=82)
```

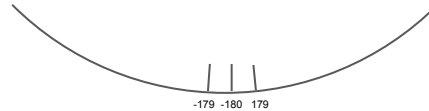
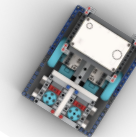
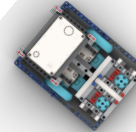
```
br.AbsTurnRightAndDriveOnHeading(heading=130, distance=70) #at the target
```

What's going on at -180 and +179?



The gyro only knows angles between (and including) -180 and +179. The angles do not add up if you keep turning more than one full circle.

This robot is on heading 130. It needs to turn to the right 90 deg. It will go past 179 and then enter the negative headings. Is that a problem? Depends on how you code it. Similarly, will you have a problem when the robot crosses 0?

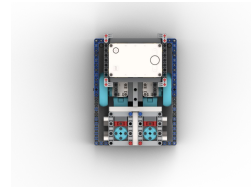
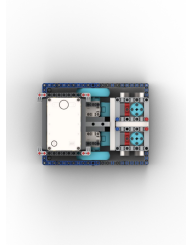
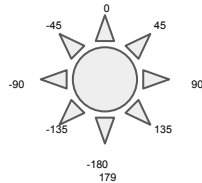


GyroDrive on headings close to -180/+179 is dangerous

Let's say you are using the gyro on heading 90. If the gyro sees a number bigger than 90, the robot turns to the left a tiny bit. And if the gyro sees a number less than 90, the robot turns to the right. No problem.

But, if we tried to GyroDrive on 179, the next heading to the right is -180, which is less than 179 (way less!). And if the actual heading is less than the desired heading, we normally turn to the right. But we are already to the right of our desired heading.

We need to prevent GyroDrive on any headings close to -180/+179



What if someone REALLY wants to drive on headings near -180/+179?

Three options:

1. Wall square and reset the gyro on a different heading so that the absolute heading isn't near -180/+179
2. Use `move_tank` or something similar (don't use the gyro)
3. Find a different way

Could there be a “RelativeTurn(relDeg)” method?

Absolutely!!

Pseudocode:

What heading am I on now? Which way am I turning?

Add relDeg (negative numbers turn to the left). This is the absolute heading that the robot needs to turn to.

Calculate the stop (absolute) heading so that the overshoot is accounted for (maybe 5 degrees early?). Think about which way the robot is turning.

Is the stop heading > 179 or < -180 ? If so, do some simple math to fix it (add or subtract 360)

Turn to the stop heading. If relDeg is negative, turn to the left. The robot should stop close to the desired new (absolute) heading