PRESENTATION 2 · LEARNING WITH C

Three targets

Position and Compass sensors, planning a route with four turns, and Session 2 step by step.

TheBotCup 2026 · AiBotCup Simulator



REMEMBER FROM PRESENTATION 1

Four ideas we use again today

Top to bottom

Every 60 ms Game1() runs; the first true branch wins.

Straight

4.5 and 4.5 = **27 cm** every second.

Spin left

−2 and 2 = about **92°** every second.

Time sensor

Whole seconds since Play. Time <= 5 = 6 seconds.

TODAY

From one turn to four

1

New sensors

Position and Compass.

2

Plan

A route that visits 1, 2, 3 in order.

3

Write & run

Ten branches in C, then Play.

4

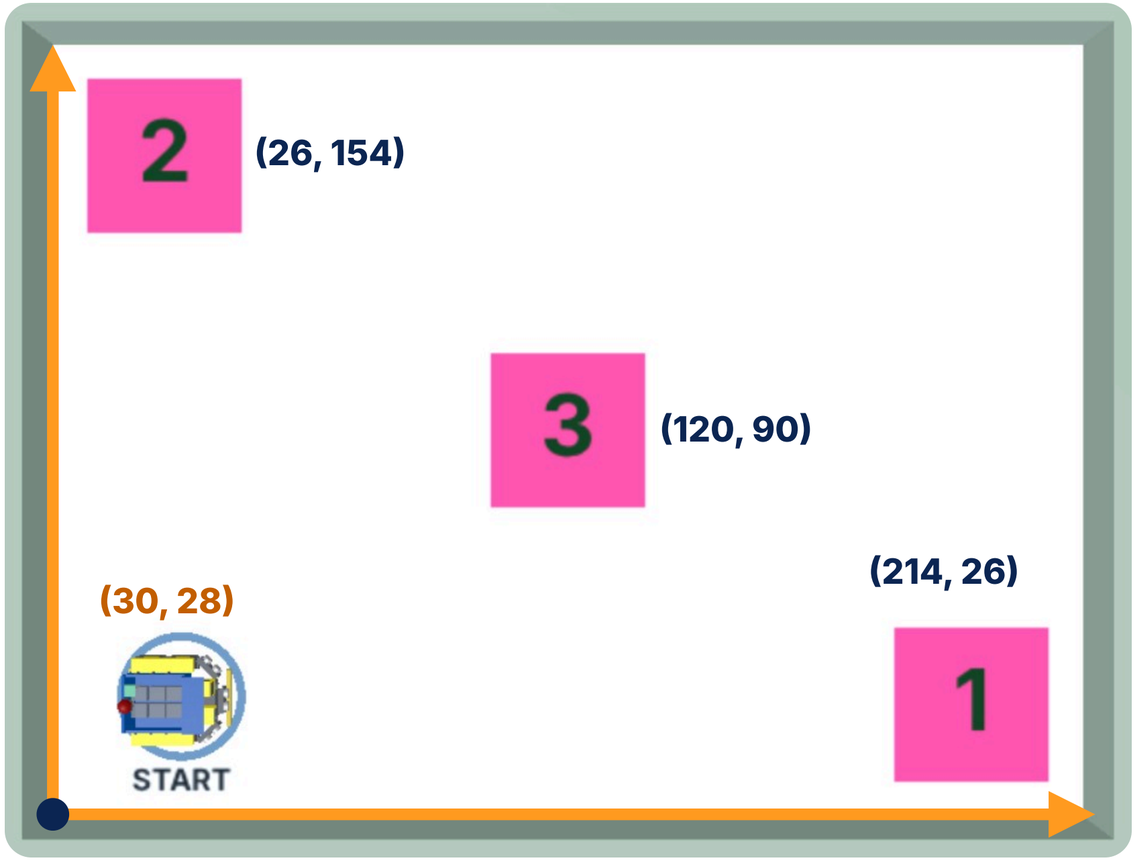
Improve

Why the path bends, and how to fix it.

PART 1

Where am I? Which way?

The Position and Compass sensors



POSITION SENSOR

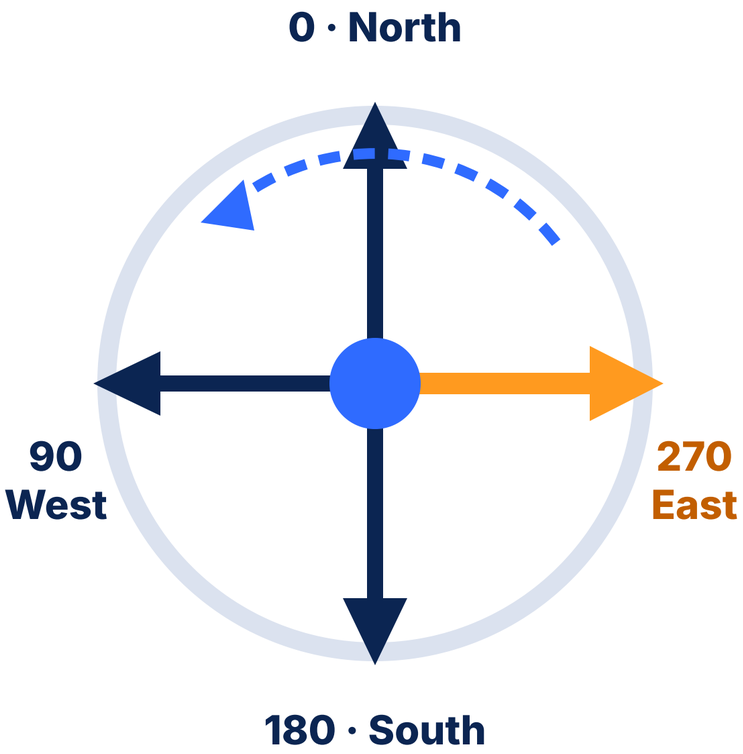
Where is the robot?

PositionX: cm from the left wall, 0 → 240.

PositionY: cm from the bottom wall, 0 → 180.

(0, 0) is the **bottom-left** corner. The robot starts at **(30, 28)**.

Target centres: **1** (214, 26) · **2** (26, 154) · **3** (120, 90). Each target is 36 × 36 cm.



COMPASS SENSOR

Which way is the robot facing?

Compass goes from 0 to 359 degrees.

Up = **0**, left = **90**, down = **180**, right = **270**.

At START the robot faces right: **270**.

Turning **left** makes the number go **up**: 270 → 359 → 0 → 90 → 180.

TRY IT

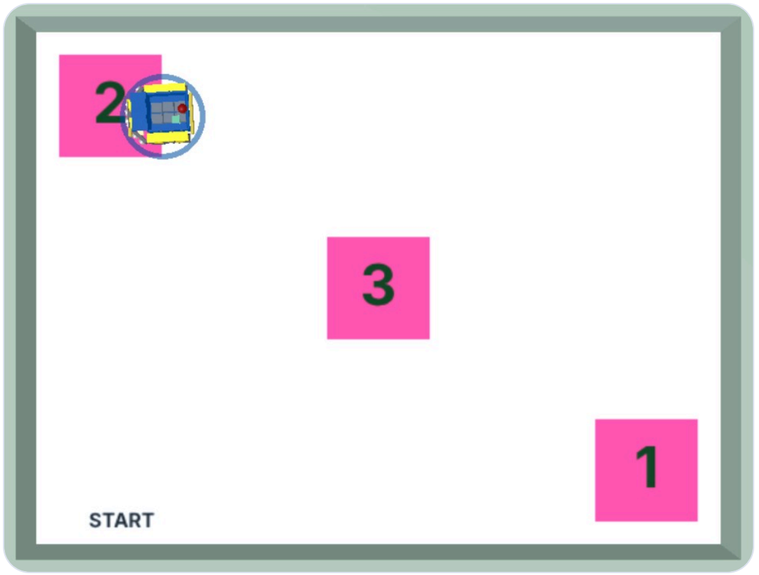
Drive by hand and read the sensors

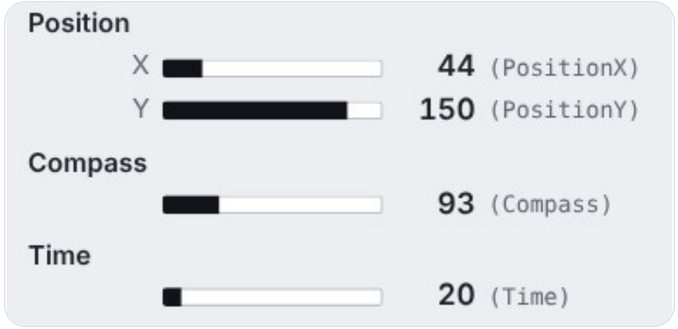
1 Open Session 2. Use the **joystick** or the arrow keys.

2 Drive onto each target. Write down **PositionX**, **PositionY** and **Compass**.

3 Press **Restart session** before running your program.

Driving by hand never counts for the session: it is for exploring.





PART 2

Plan the route

Look · plan · calculate · timeline

STEP 1 · LOOK

The mission

2

Session 2 · Three targets

Visit target 1 (bottom right), then target 2 (top left), then target 3 (centre), in that order.

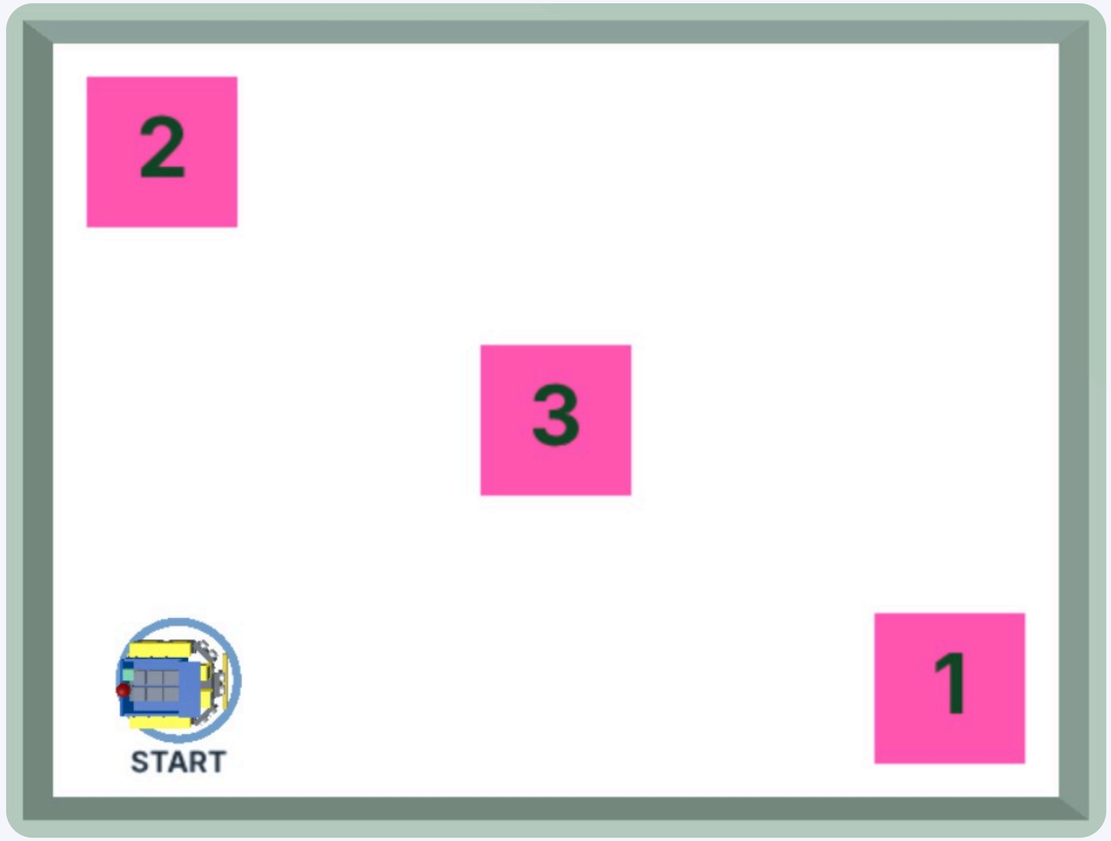
Target 1: +50. Target 2: +50. Target 3: +100. Only the next target in order counts. Time limit 4:00.

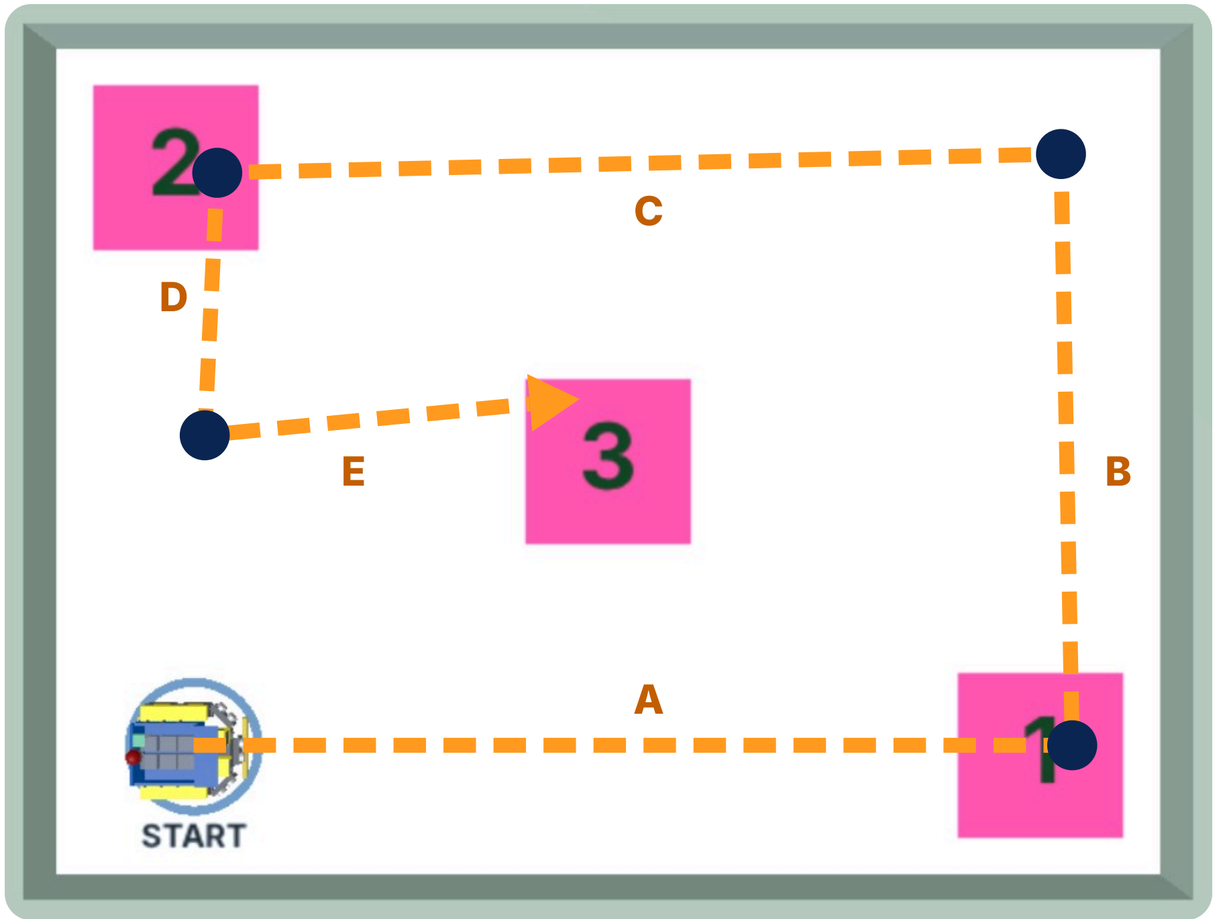
Not done yet

Visit **1 → 2 → 3** in that order. Only the next one counts.

Points: **+50**, **+50**, **+100** = 200, within **4:00**.

Same field as Session 1, with a wall around it.





STEP 2 · PLAN

Five straight legs, four left turns

A right, through **target 1**

B up the right side

C left, into **target 2**

D down a little

E right, into **target 3**

Going around the edge keeps us away from target 3 until the end.

STEP 3 · CALCULATE

How long is each leg?

Leg	Where to	Seconds × 27 cm/s	Ends near
A	X 30 → inside target 1 (196–232)	7 s = 189 cm	X ≈ 220
B	Y 28 → near the top	5 s = 135 cm	Y ≈ 160
C	X 220 → inside target 2 (8–44)	7 s = 189 cm	X ≈ 38
D	Y 150 → level with target 3 (72–108)	2 s = 54 cm	Y ≈ 95
E	X 31 → inside target 3 (102–138)	about 3 s = 81 cm	X ≈ 110

Each turn: **−2 and 2 for 1 second** ≈ a quarter turn left.

STEP 4 · TIMELINE

The whole trip in 30 seconds



Blue = straight

4.5 and 4.5

Orange = turn left

−2 and 2, one second

No gaps

Each range starts right after the last: one else if chain.

PART 3

Write and run

Ten branches · Build · Load · Play

STEP 5 · WRITE

The whole program: ten branches

```
// Session 2 · Three targets

void Game1() {
    if (Time <= 6) {                        // Forward1 · A
        WheelLeft = 4.5; WheelRight = 4.5;
    } else if (Time == 7) {                 // Turn1
        WheelLeft = -2; WheelRight = 2;
    } else if (Time <= 12) {                // Forward2 · B
        WheelLeft = 4.5; WheelRight = 4.5;
    } else if (Time == 13) {                // Turn2
        WheelLeft = -2; WheelRight = 2;
    } else if (Time <= 20) {                // Forward3 · C
        WheelLeft = 4.5; WheelRight = 4.5;
    } else if (Time == 21) {                // Turn3
        WheelLeft = -2; WheelRight = 2;
    } else if (Time <= 23) {                // Forward4 · D
        WheelLeft = 4.5; WheelRight = 4.5;
    } else if (Time == 24) {                // Turn4
        WheelLeft = -2; WheelRight = 2;
    } else if (Time <= 29) {                // Forward5 · E
        WheelLeft = 4.5; WheelRight = 4.5;
    } else {                                // Stop
        WheelLeft = 0; WheelRight = 0;
    }
}
```

Forward legs: **Time <= ...** → 4.5 and 4.5

Turns: **Time == 7, 13, 21, 24** → −2 and 2

Stop: the else, after Time 29

Each else if starts where the last one ended. No gaps.

STEP 5 · CHECK

Read it like the robot: from the top

At Time	First true branch	Wheels
0 – 6	Forward1 · A · Time <= 6	4.5 · 4.5
7	Turn1 · Time == 7	−2 · 2
8 – 12	Forward2 · B · Time <= 12	4.5 · 4.5
13	Turn2 · Time == 13	−2 · 2
14 – 20	Forward3 · C · Time <= 20	4.5 · 4.5
21	Turn3 · Time == 21	−2 · 2
22 – 23	Forward4 · D · Time <= 23	4.5 · 4.5
24	Turn4 · Time == 24	−2 · 2
25 – 29	Forward5 · E · Time <= 29	4.5 · 4.5
30 +	Stop · else	0 · 0

WORK FASTER

Copy, paste, change

1 Type Forward1 and Turn1 first: two lines each.

2 Select both, **Ctrl+C**, put the cursor below, **Ctrl+V**.

3 In the copy change only the **Time** number and the comment.

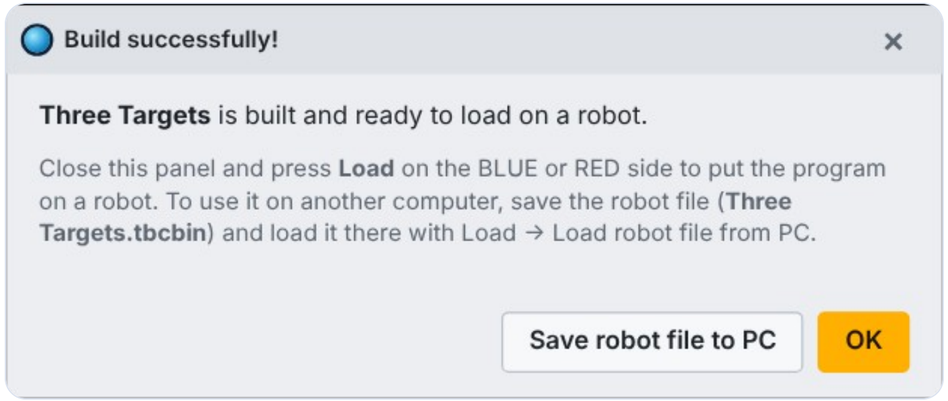
4 The first copied if becomes else if: only the top branch is **if**.

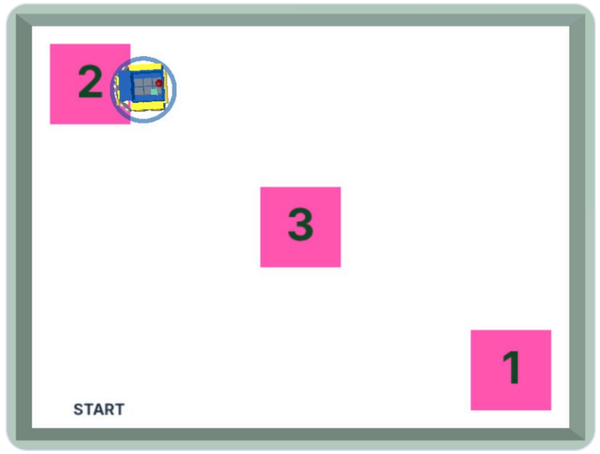
C: copy the **} else if (...) {** lines together with their bodies.

STEP 6 · TEST

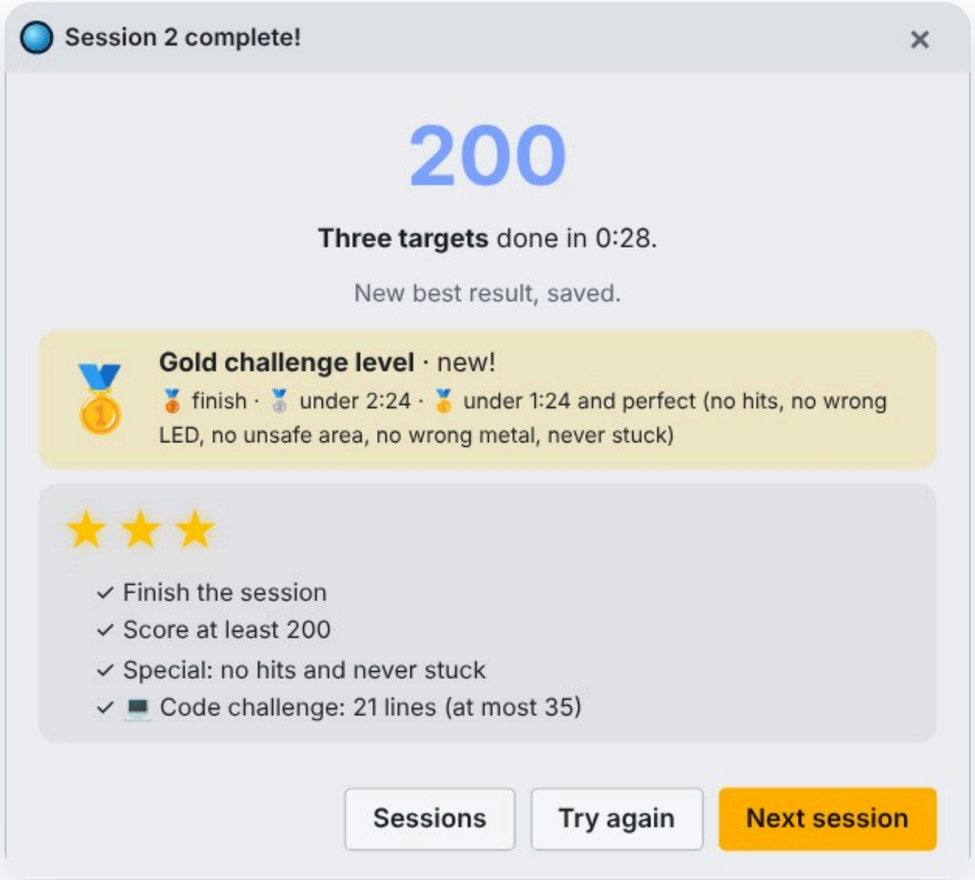
Build, Load on BLUE, Play

1 Build → OK → **2 Save to PC** (keep a copy) → **3** BLUE **Load** → Three Targets → **4 ▶ Play**





At **Time 20** the robot reaches target 2. The feed says **Target 1 reached +50**.



RESULT · TESTED

200 points in 0:28 · Gold

Medals · Bronze: finish · Silver: under 2:24 · Gold: under 1:24 and perfect.

Stars · finish · score 200 · no hits and never stuck.

Code challenge · at most 35 lines. We used 21.

PART 4

Why does the path bend?

Small errors add up

CHECK WITH THE COMPASS

Each turn is a little more than 90°

After	We wanted	Compass read	Off by
Turn1	0	**2**	2°
Turn2	90	**93**	3°
Turn3	180	**185**	5°
Turn4	270	**277**	7°

−2 and 2 for 1 second = **91.7°**, not 90°.

The extra bit **adds up** after every turn.

Leg E climbs: target 3 was reached at **Y ≈ 103**. Its top edge is **108**.

IMPROVE

Fine-tune the turn, then re-time

CHANGE 1

Smaller turn

All four turn branches: WheelLeft **−2**, WheelRight **1.9** ≈ **89°** a second.

CHANGE 2

Longer leg D

Forward4 **22–24** · Turn4 **25** · Forward5 **26–30**.

Tested: target 3 is reached at **Y ≈ 80**, near its middle, in **0:29**. Later we will use the Compass in conditions to stop each turn at exactly the right heading.

DID NOT WORK?

Find the leg, change one thing

What you see	What to do
Build error with a line number	; after every line, every { closed by }
No “Target 1 reached”	Forward1: **Time <= 6** → 5 or 7
Hits the top wall on leg B	Forward2: **Time <= 12** → 11
“Target 2 first”	The robot crossed target 3 too early: check the turn Times
Misses target 3 above or below	Forward4 Time, or the turn wheels (−2, 1.9)
Stops before target 3	Forward5: make the 29 bigger

TRY IT YOURSELF

Three challenges

Shorter code

Write a function forward() and use it in every leg. How many lines now?

Straighter

Use the fine-tuned turns and check the Compass after each one.

Next session

Presentation 3 · **Avoid the walls** with the ultrasonic sensors.