PRESENTATION 17 · LEARNING WITH C

Timer

No position, no walls to see: drive for the right number of seconds, then stop.

TheBotCup 2026 · AiBotCup Simulator



RECAP

Conditions we have used

Ultrasonic
how far the walls are

Colour
what is under the robot

Position
where the robot is

Time
new today

One robot variable we have not used yet: **Time**.

PART 1

The mission: Session 17

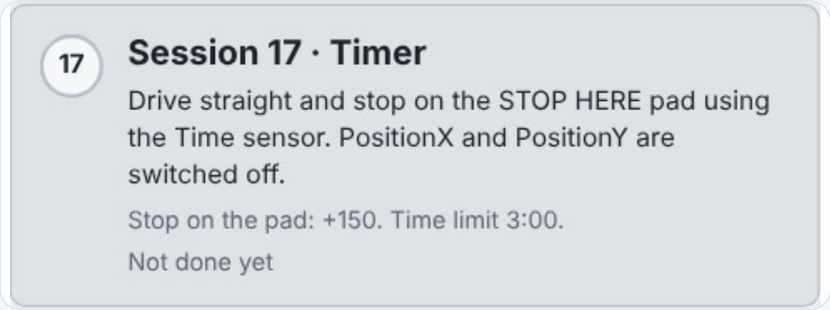
Stop on the pad, by time

SESSION 17 · TIMER

Drive straight, stop on STOP HERE

Use the **Time** sensor. Position X and Y are **switched off**.

On the pad: **+150**. Time limit **3:00**. Code challenge: at most **20** lines.





THE FIELD

One straight line

START bottom centre, facing **up**.

The pink **STOP HERE** pad is straight ahead.

Nothing to see on the way: **only time** helps.

PART 2

Time as a condition

Seconds since Play

THE IDEA · THE TIME SENSOR

A clock you can test

Time

counts whole **seconds** while the robot runs: 0, 1, 2, 3…

In a condition

Time <= 4 = “during the first five seconds” (0, 1, 2, 3, 4).

Same speed every time + same seconds = **same distance**.

THE PLAN

Go while time is small, then Stop

1 Go

Time ≤ 4 → drive



2 Stop

the else → wheels 0

Top to bottom, the first true condition wins: Go while Time ≤ 4, then Stop takes over.

PART 3

Step by step in C

Just two branches

STEP 1 · THE WHOLE PROGRAM

The shortest program so far

Branch	Condition	Wheels
1 Go	Time ≤ 4	4.5 / 4.5
2 Stop	the else	0 / 0

The number to find is the **4** in Time ≤ 4. We tested **4**.

STEP 2 · LINES 1–9

Go and Stop

```
1  // Session 17 · Timer
2
3  void Game1() {
4      if (Time <= 4) {                  // Go
5          WheelLeft = 4.5; WheelRight = 4.5;
6      } else {                          // Stop
7          WheelLeft = 0; WheelRight = 0;
8      }
9  }
```

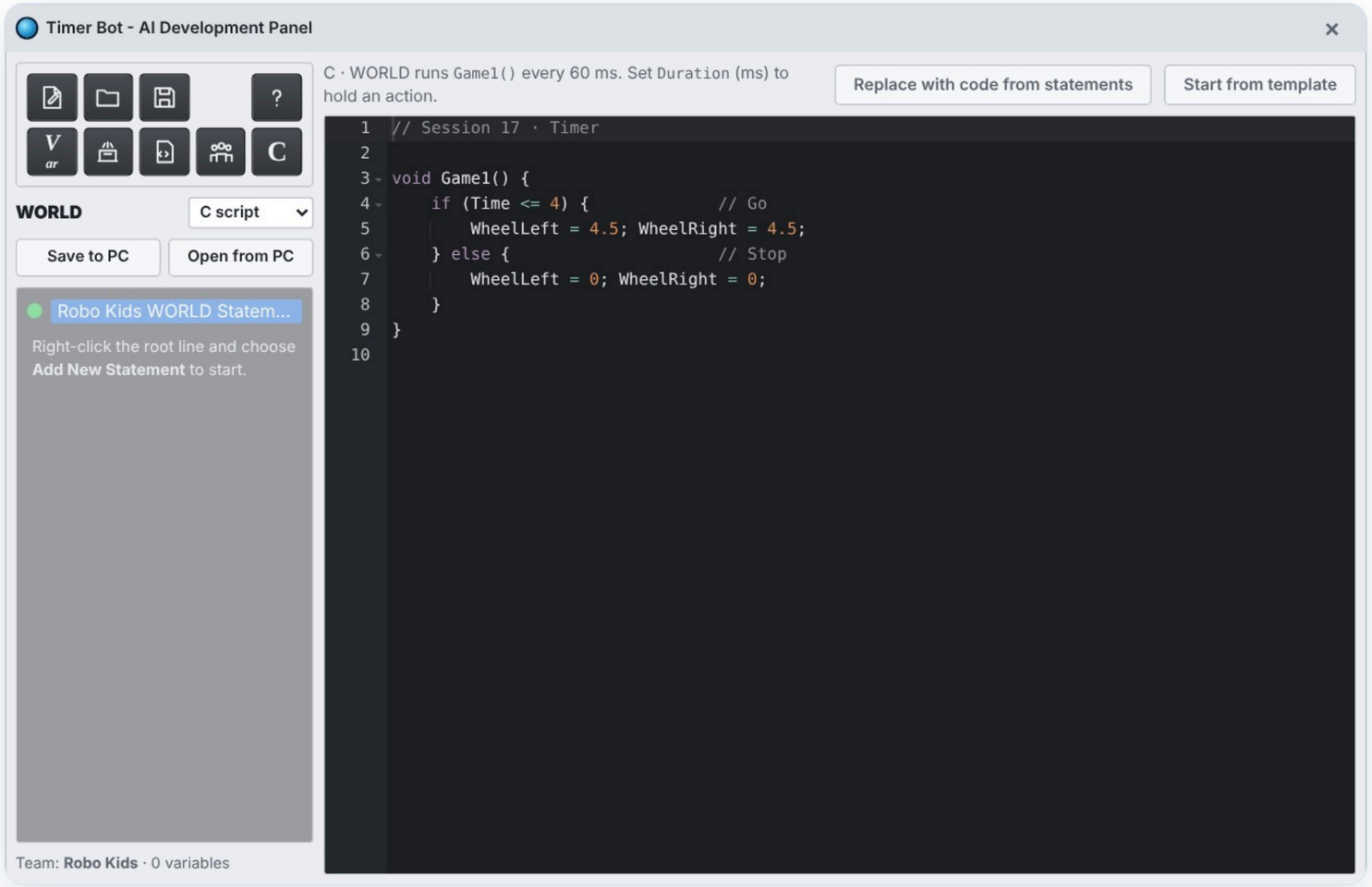
Go: Time ≤ 4 → 4.5 / 4.5. Time counts whole seconds.

Stop: the else → 0 / 0, for the rest of the run.

STEP 3 · BUILD AND LOAD

The whole program, built

1. WORLD menu: **C script**.
2. Type the program, name it **Timer Bot**.
3. Press **Build**, then **OK**.
4. Press **Load** on the BLUE side.





STEP 4 · RUN · EARLY

Go is running

Position

X 0 (PositionX)

Y 0 (PositionY)

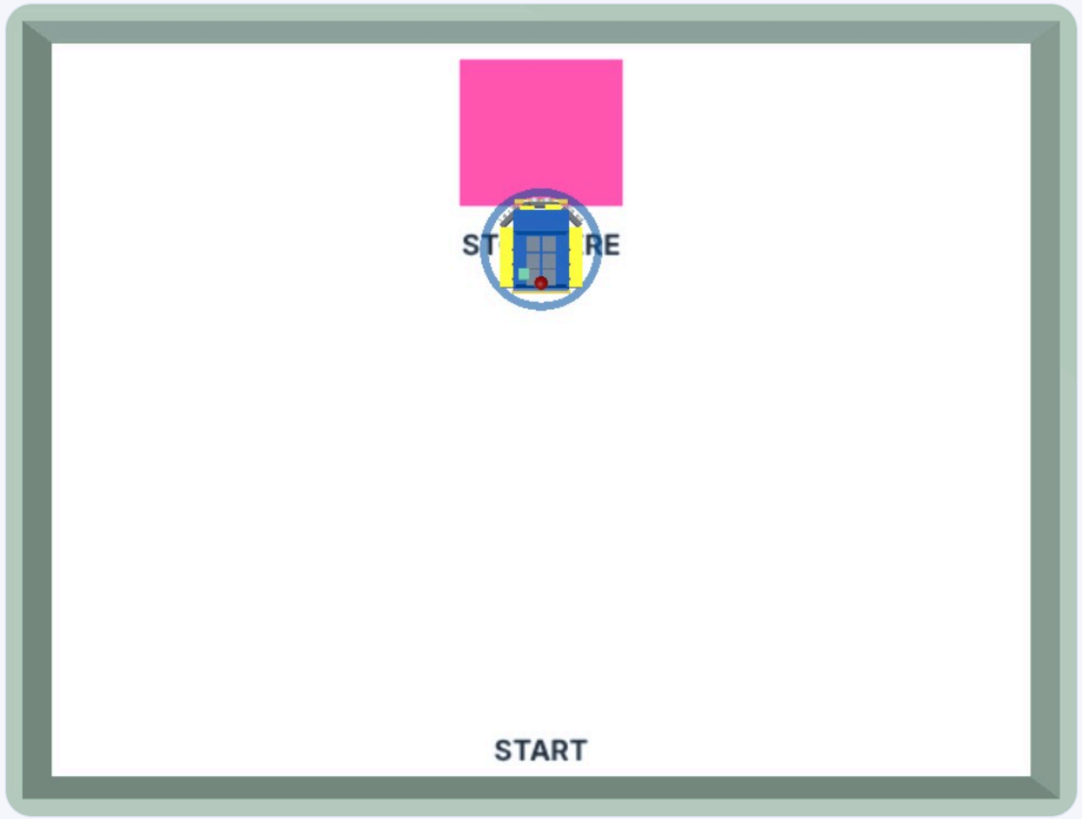
Compass

0 (Compass)

Time

1 (Time)

Position reads **0, 0**: switched off. Time **1** ≤ 4: **Go**.



STEP 5 · RUN · 0:04

At the pad

Position

X 0 (PositionX)

Y 0 (PositionY)

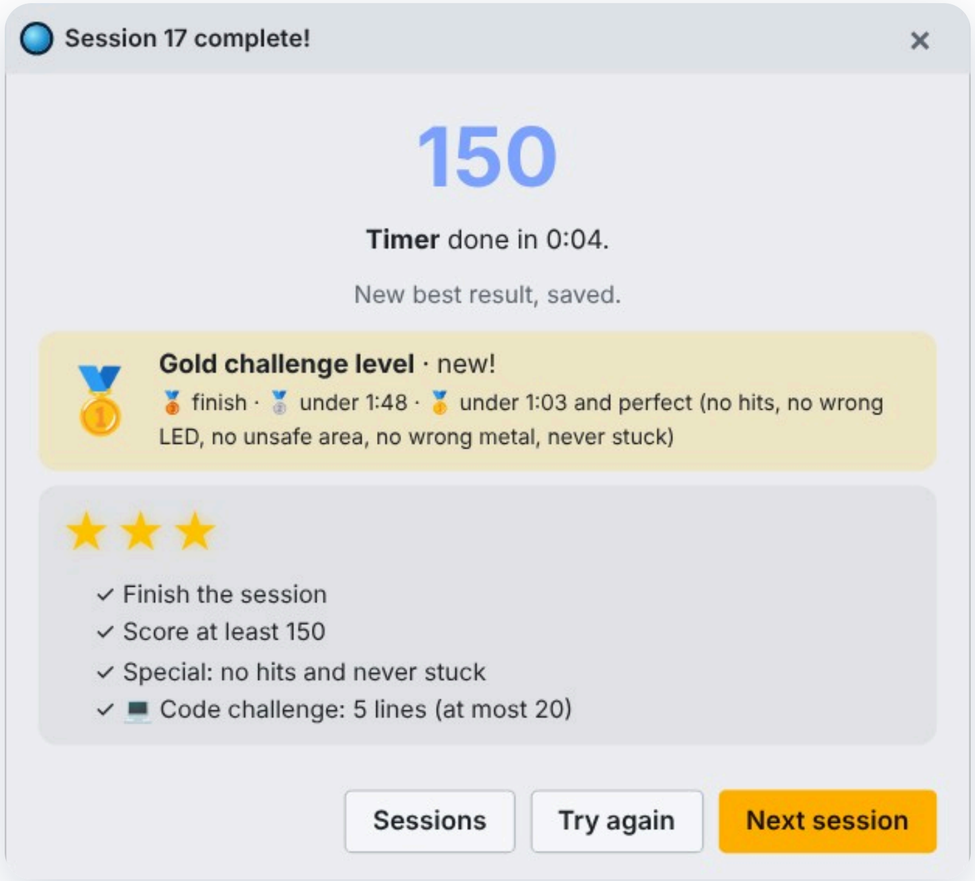
Compass

0 (Compass)

Time

4 (Time)

Time **4**: the robot reaches the pad and the session completes.



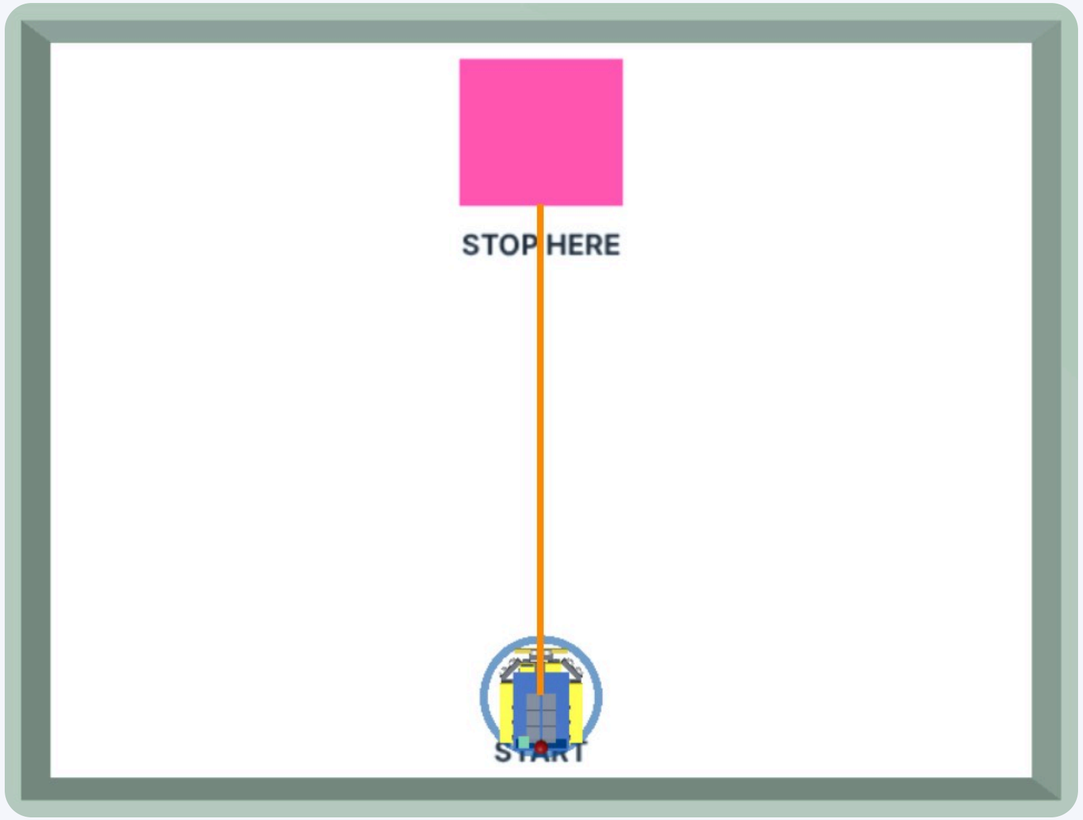
RESULT · TESTED

150 points in 0:04 · Gold

Medals · Silver: under 1:48 · Gold: under 1:03 and perfect.

Stars · finish · 150 · no hits and never stuck.

Code challenge · at most 20 lines. We used 5.



THE PATH · RECORDED

One straight line

START to the pad, nothing else.

When nothing can be seen, **time** is a sensor too.

DID NOT WORK?

Check first, then tune carefully

What you see	What to do
Stops before the pad	Change 4 to 5 in Time <= 4
Drives past the pad	Change 4 to 3 in Time <= 4
Never stops	The else **Stop** needs wheels 0 / 0
Old program still on the robot	**Build** again, then **Load** on BLUE

NEXT TIME

Presentation 18 · Search and return

Find two metals and bring them back to **the deposit box** you started in.