Self-driving car project
Week 8 Exercises

Programming robots in TINKERCAD

This week you be using TinkerCad Circuits to continue programming the self-driving car. This workshop aims to revisit the text programming done previously and continue to develop your text programming skills.

If you didn't complete the circuit from Week 7, copy and tinker the "Elegoo robot car" from the classroom activities then complete Week 7 silver before starting these exercises.

Please complete the bronze requirements before starting the silver, and complete the silver before starting the gold.

Joining the Classroom

Go to **www.tinkercad.com/joinclass**

Enter class code: Y81PSFC2Y

Enter your nickname as provided by us

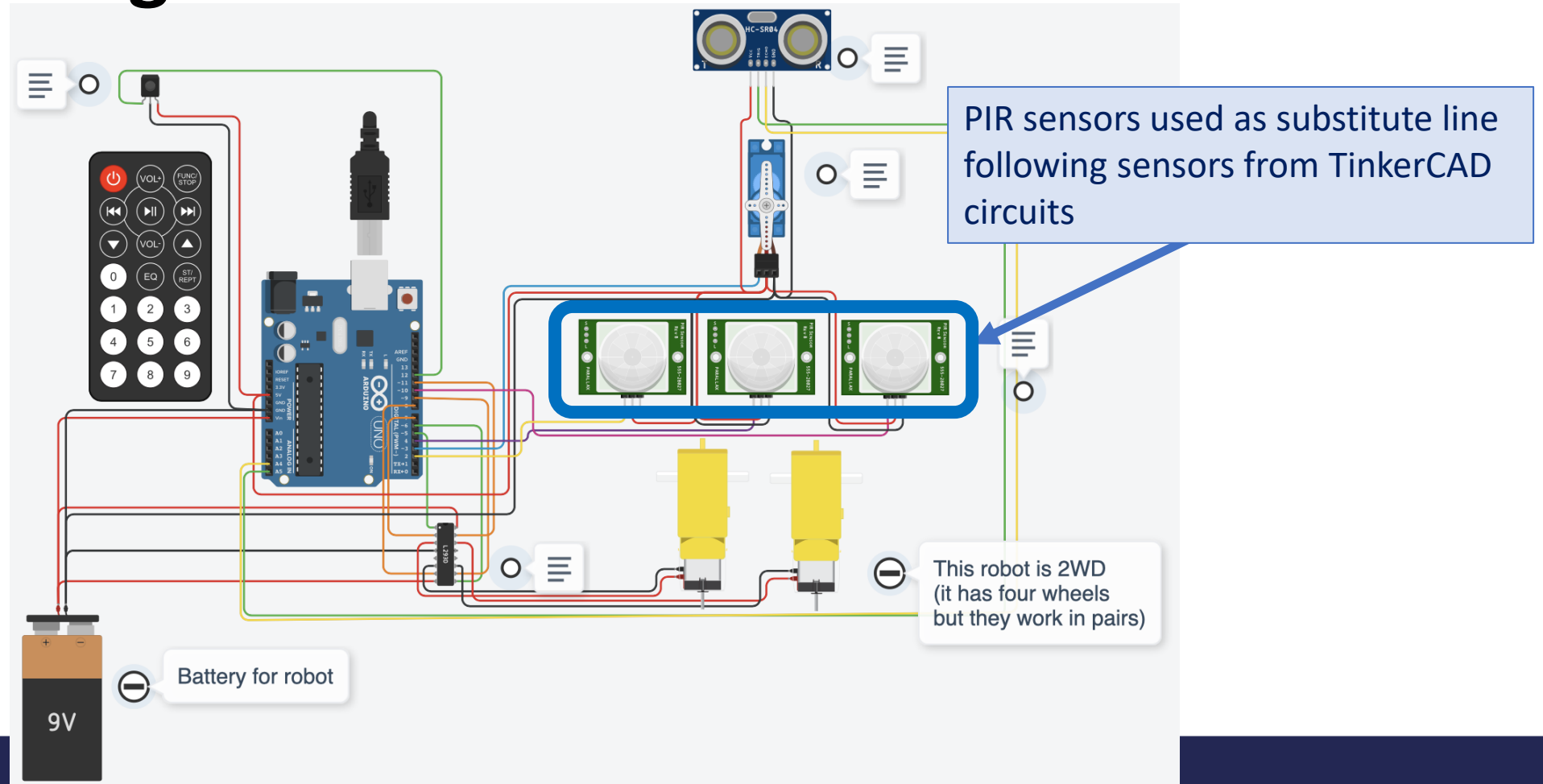You should then continue with the Elegoo circuit from Week 7

```
void robotReverse() {
  //add code here
}

void robotLeft() {
  //add code here
}

void robotRight() {
  //add code here
}

void robotStop() {
  //add code here
}
```

Line following sensors

PIR sensors used as substitute line following sensors from TinkerCAD circuits

This robot is 2WD (it has four wheels but they work in pairs)

Battery for robot

BRONZE Challenge:

Tasks:

1. Add #define statements for the 'line following sensor' pins (see notes for details)
2. Set pinModes for the line following sensors as Inputs
3. Move code written last week from the loop function to new functions
4. Add a function called `void lineFollow()' then call this function from the loop. This new function will be developed in the next task

Notes:

- Line sensor pins:
  - left line pin: 2
  - mid line pin: 4
  - right line pin: 10
- Functions should represent 'actions' we may want to easily repeat.
  e.g. slowly accelerating might have a function:
  - void drivePattern() {...}
  - void lookAround() {...}
  - void slowAccelerate(int targetSpeed){... }
  - void obstacleAvoid(){...}

SILVER Challenge:

Tasks:

1. Add 3 boolean (bool) variables to the lineFollow function and initialise them with values from digitalRead for each of the line sensors.
2. Print the values from the line sensors to the serial monitor (e.g. see notes)
3. Add three if-else statements of the form
   - if(left && !mid && !right)
   filling in appropriate behaviour for each

Notes:

- Defining boolean variables:
  - bool variableName;

- Reading from the sensor:
  - variableName = digitalRead(leftLinePin);

- Printing values
  - Serial.print("left: ");
  - Serial.println(left);

GOLD Challenge:

Aims:

- Consider what happens if two of the sensors are detecting the line?
- Add appropriate conditions and behaviours to match
- You may wish to consider the rate of turning. Do you need to add/modify driving functions elsewhere?

Notes:

- You may want to use multiple if statements to create the 'logic' of your program. E.g. if (centerDistance < 20) {
- Use the serial monitor to help solve problems by printing out what part of the code is running e.g. Serial.println("driving forward"); or Serial.println("found obstacle");
- Remember to watch the video to guide you through this task.

Extension Challenge:

What else can you add to this program?

- Line sensors can be used for functionality other than following a line. Can you make your autonomous count how many lines it has crossed with all 3 sensors detecting the line, e.g. driving over a bar code.

- After driving over 3 lines, the robot should perform a behaviour.