

IR Raw Values

```
code.py
1 import time
2 from adafruit_crickit import crickit
3
4 ss = crickit.seesaw
5
6 # Define pin numbers
7 ir_1 = crickit.SIGNAL1 # Right sensor
8 ir_2 = crickit.SIGNAL8 # Left sensor
9
10 # Set pin modes (configure pins as input with pull-up resistors)
11 ss.pin_mode(ir_1, ss.INPUT_PULLUP)
12 ss.pin_mode(ir_2, ss.INPUT_PULLUP)
13
14 while True:
15     # Read sensors
16     right_sensor = ss.digital_read(ir_1)
17     left_sensor = ss.digital_read(ir_2)
18     print("right", right_sensor, "left", left_sensor)
19     time.sleep(0.5)
```

Code Explanation:

```
import time
from adafruit_crickit import crickit
ss = crickit.seesaw
```

- `time` → used for delays (`time.sleep()`).
- `adafruit_crickit` → **library for controlling motors, sensors, and signals on Crickit.**
- `ss` = shortcut to access Crickit's **Seesaw chip** (handles sensors and signals).

```
ir_1 = crickit.SIGNAL1 # Right sensor
ir_2 = crickit.SIGNAL8 # Left sensor
```

You are connecting two IR sensors (infrared obstacle or line sensors).

- `SIGNAL1` and `SIGNAL8` are Crickit's signal pins.
- Here:
- `ir_1` = right IR sensor

- `ir_2` = left IR sensor

```
ss.pin_mode(ir_1, ss.INPUT_PULLUP)
ss.pin_mode(ir_2, ss.INPUT_PULLUP)
```

Configures both pins as inputs.

- `INPUT_PULLUP` means an **internal resistor** is enabled → the pin normally reads **HIGH (1)**, but will go **LOW (0)** when the sensor pulls it down (e.g., when it detects something)

```
while True:
```

- Keeps running forever (until stopped).

```
right_sensor = ss.digital_read(ir_1)
left_sensor = ss.digital_read(ir_2)
print("right", right_sensor, "left", left_sensor)
time.sleep(0.5)
```

Reads digital values (0 or 1) from each IR sensor.

- 1 (HIGH) = No detection (depends on sensor type).
- 0 (LOW) = Detection (object or line detected).
- Displays the readings from **both sensors** on the serial monitor.
- Waits half a second before reading again → avoids flooding output with too many prints.