

AUTOMATIC WINDOW DOOR OPENER USING TEMPERATURE SENSOR

The installation width of window frame
should be greater than 2cm(0.79in)



Automatic Window Opener

This presentation introduces the M-Ruby language and the R-Board development platform. We'll explore a practical project that uses these technologies to create a buzzer alert system that sounds when a window is opened or closed. We'll also discuss the steps involved in setting up the environment, connecting the hardware, writing the M-Ruby code, and integrating the buzzer with the window sensors.

Connecting the R-Board to the System

1

Connect R-Board

Connect the R-Board to the computer or device you'll be using for development and programming.

2

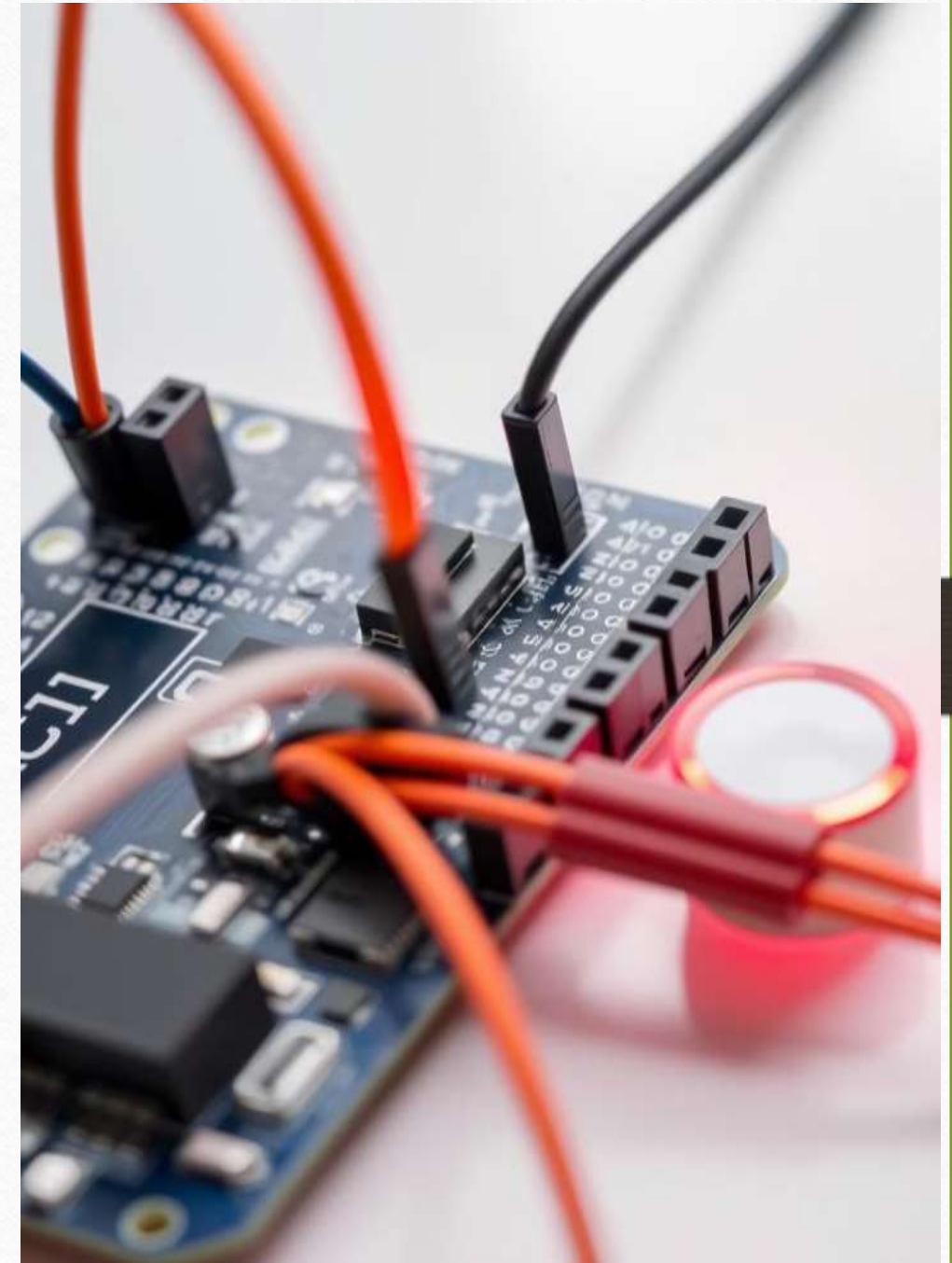
Attach Temperature Sensor

Connect the window sensor to the R-Board. This sensor will detect whether the window is open or closed.

3

Connect Buzzer

Attach the buzzer to the R-Board. This will emit the sound when a window event occurs.



Coding the Window Open/Close Detection

1 Read Sensor Data

Write M-Ruby code to read data from the window sensor. The sensor will provide a signal indicating the window's state (open or closed).

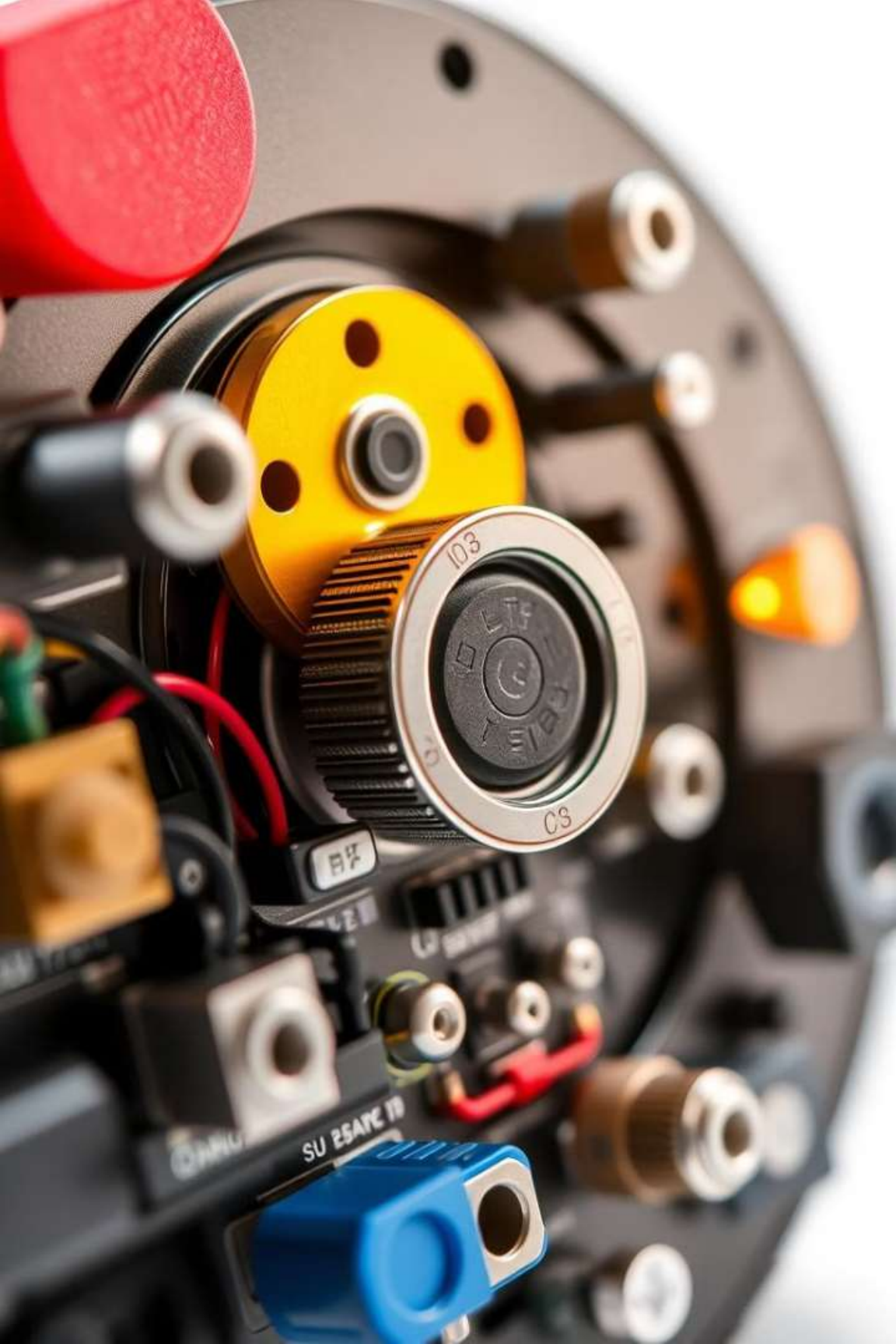
3 Trigger Events

Trigger events based on the processed sensor data. For example, if the window changes from closed to open, initiate the buzzer sound.

2 Process Sensor Data

Process the sensor data to determine whether the window has changed state (from open to closed or closed to open).

```
Code Costumes Sounds Ruby
1 BUZZ_PIN = 15
2 ADC_PIN = 20
3 THRESHOLD = 25
4 SCALE_FACTOR = 0.0195
5 OFFSET = 0.4
6
7 # Initialize GPIO and ADC
8 led = GPIO.new(BUZZ_PIN)
9 adc = ADC.new(ADC_PIN)
10 led.setmode(0)
11
12 # Initialize flag
13 window_open = false
14
15 loop do
16   # Read and process ADC value
17   sensor_value = (adc.read - OFFSET) / SCALE_FACTOR
18   puts(sensor_value)
19
20   # Check if temperature exceeds threshold and window is not open
21   if sensor_value >= THRESHOLD && !window_open
22     led.write(1)
23     puts("Window is open")
24     window_open = true
25   elsif sensor_value < THRESHOLD && window_open
26     led.write(0)
27     puts("Window is close")
28     window_open = false
29   end
30
31   sleep(1)
32 end
```

Generating the Buzzer Sound

Buzzer Control

Write M-Ruby code to control the buzzer. The code will send signals to the buzzer to activate and deactivate the sound.

Sound Duration

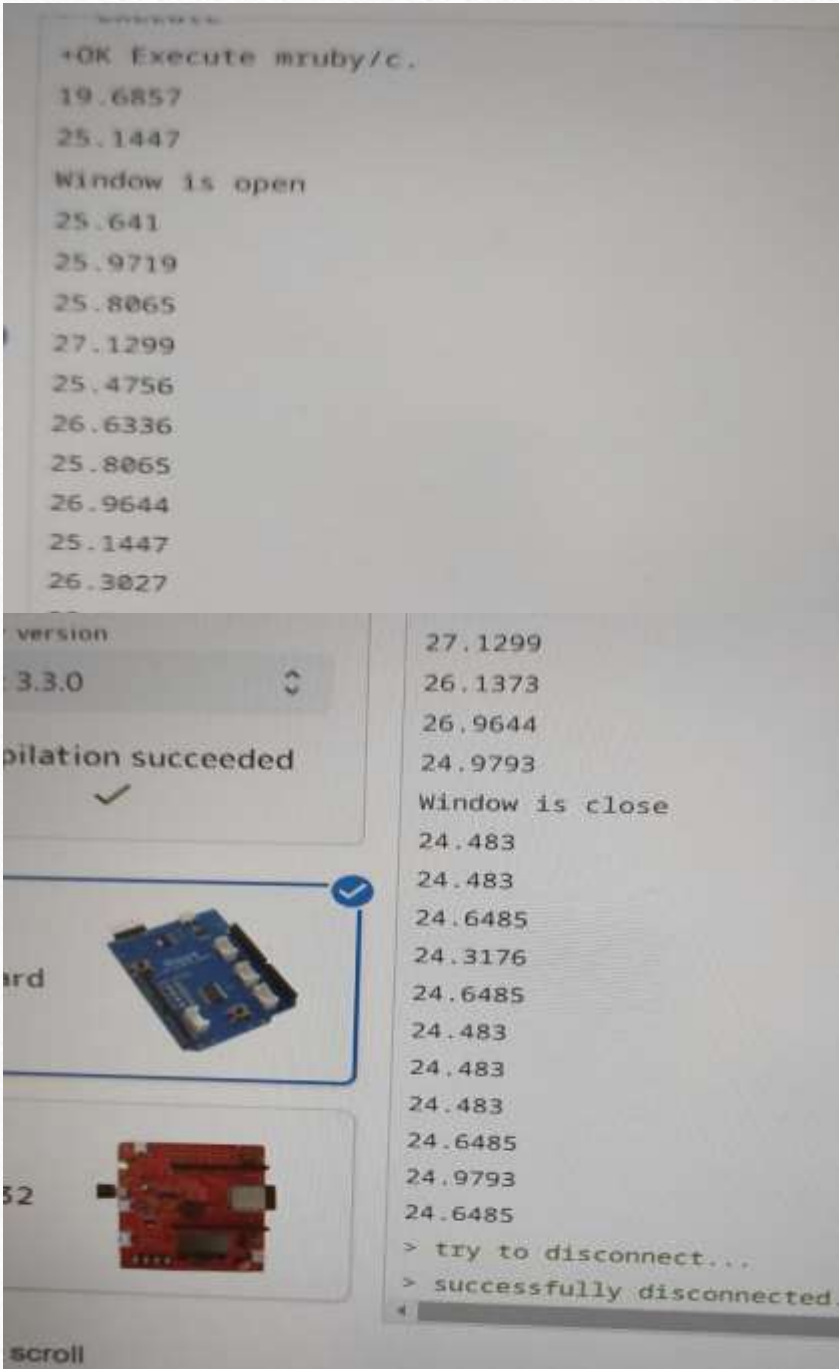
Set the duration of the buzzer sound. This could be a short beep or a longer, continuous sound depending on the application's needs.

Sound Frequency

Adjust the frequency of the buzzer sound to create a specific tone or pitch. This can be customized for different alerts.

OUTPUT

Window Event	Sensor Data	M-Ruby Code	
Window Opened	Sensor Change	Trigger Buzzer On	Buzzer Sounds
Window Closed	Sensor Change	Trigger Buzzer Off	Buzzer Silence





Thank You

Feeling gratitude and not expressing it is like
wrapping a present and not giving it.