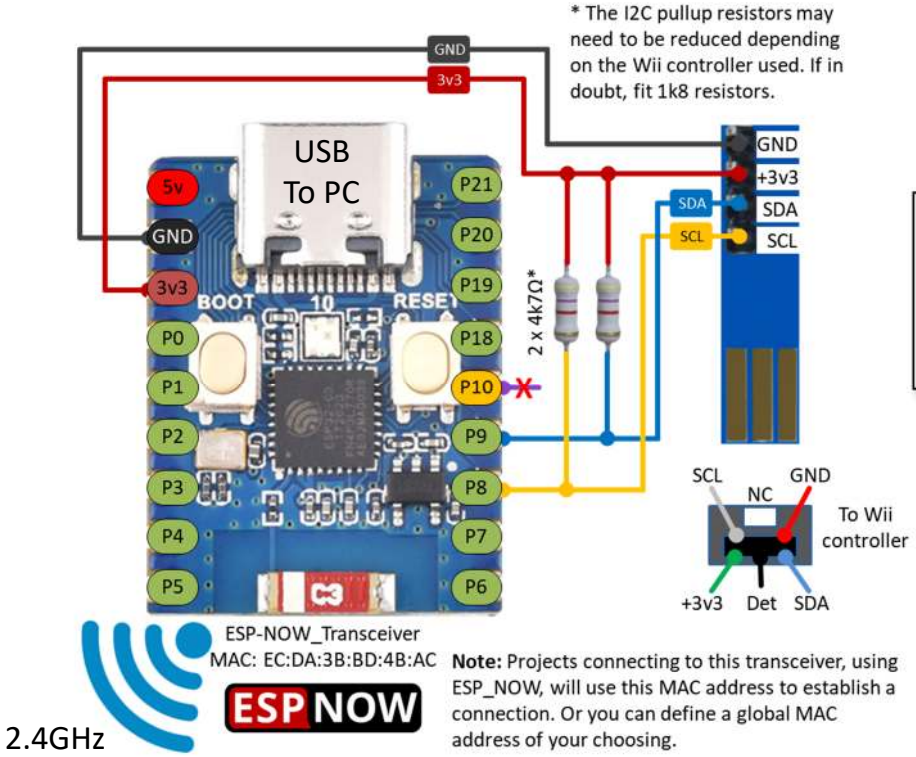
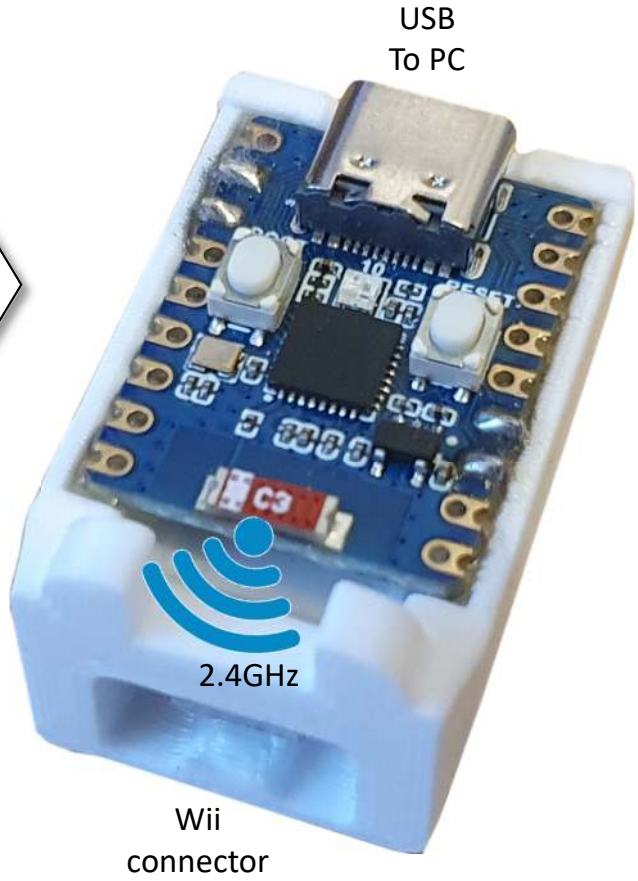
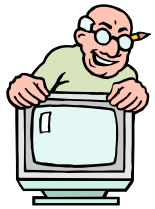
Wii ESP32-C3 Transceiver

Circuits & Wiring





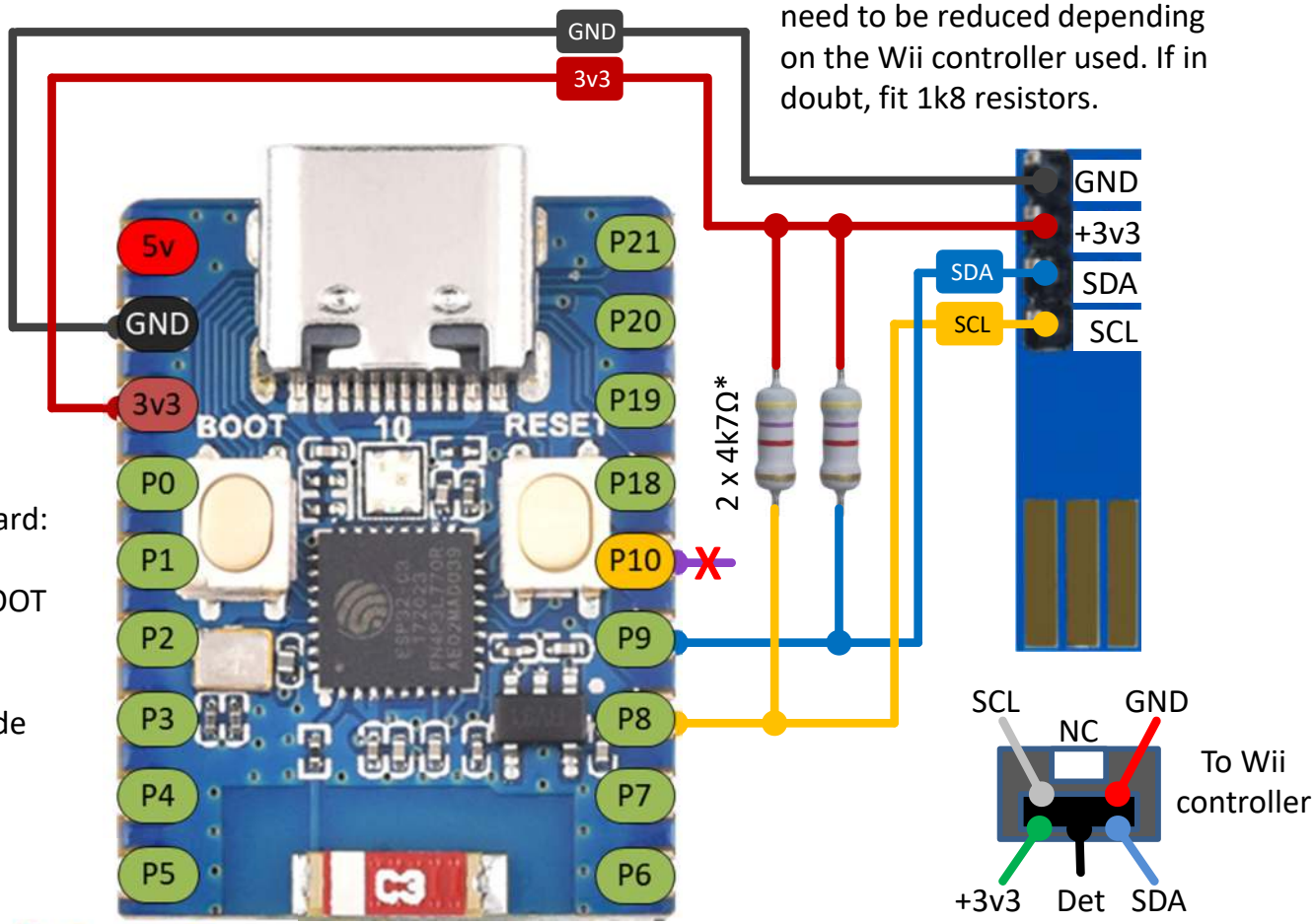


A simple low cost 2.4GHz transceiver, for use with many of my projects.

Wii Transceiver ESP32-C3 Zero

Note GPIO10 is used by the RGB LED on this board.

* The I2C pullup resistors may need to be reduced depending on the Wii controller used. If in doubt, fit 1k8 resistors.



IMPORTANT

To flash this C3-Zero board:

- Press & hold RESET
- Then press & hold BOOT
- Then release RESET
- Then release BOOT
- Then upload your code

ESP-NOW_Transceiver
MAC: EC:DA:3B:BD:4B:AC

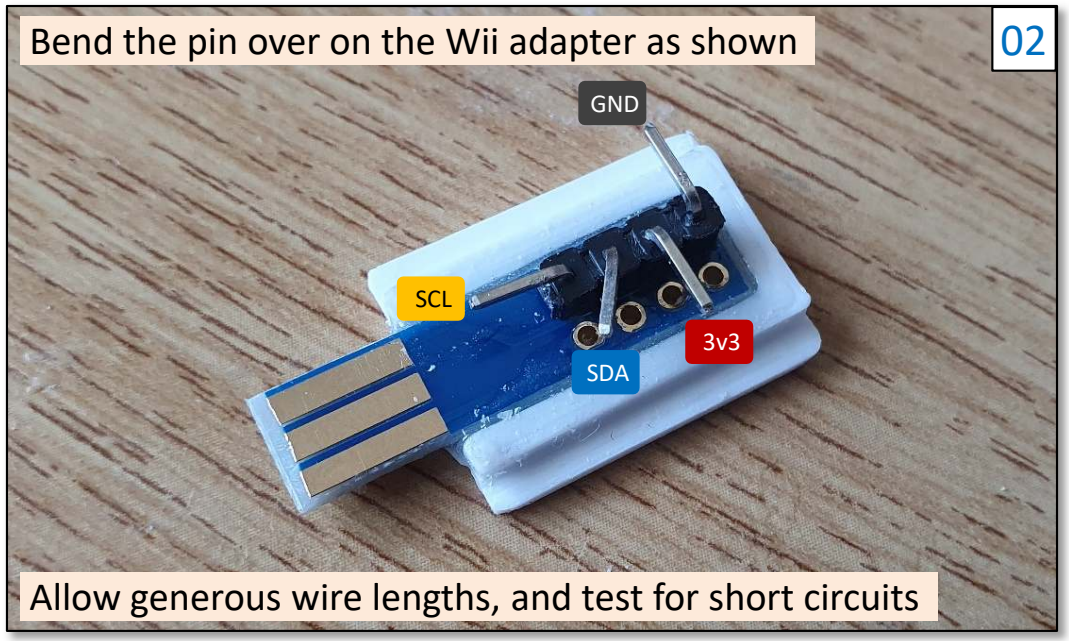
Note: Projects connecting to this transceiver, using ESP_NOW, will use this MAC address to establish a connection. Or you can define a global MAC address of your choosing.

Build Images

01 Glue the Wii adapter into the mounting plate.

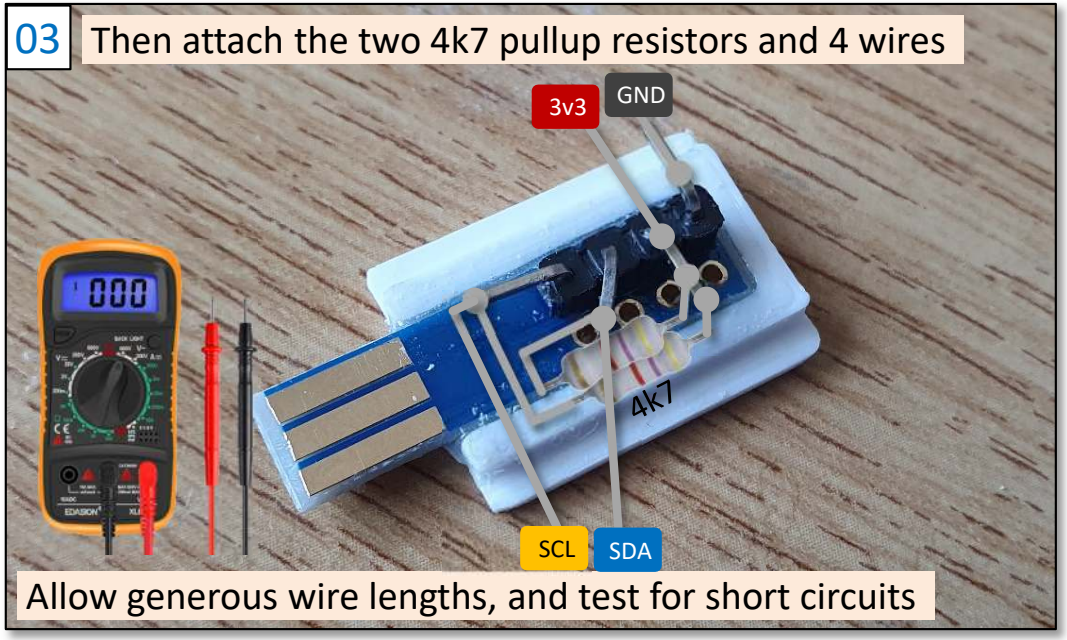
Avoid getting any glue on the pcb contacts.

Bend the pin over on the Wii adapter as shown 02



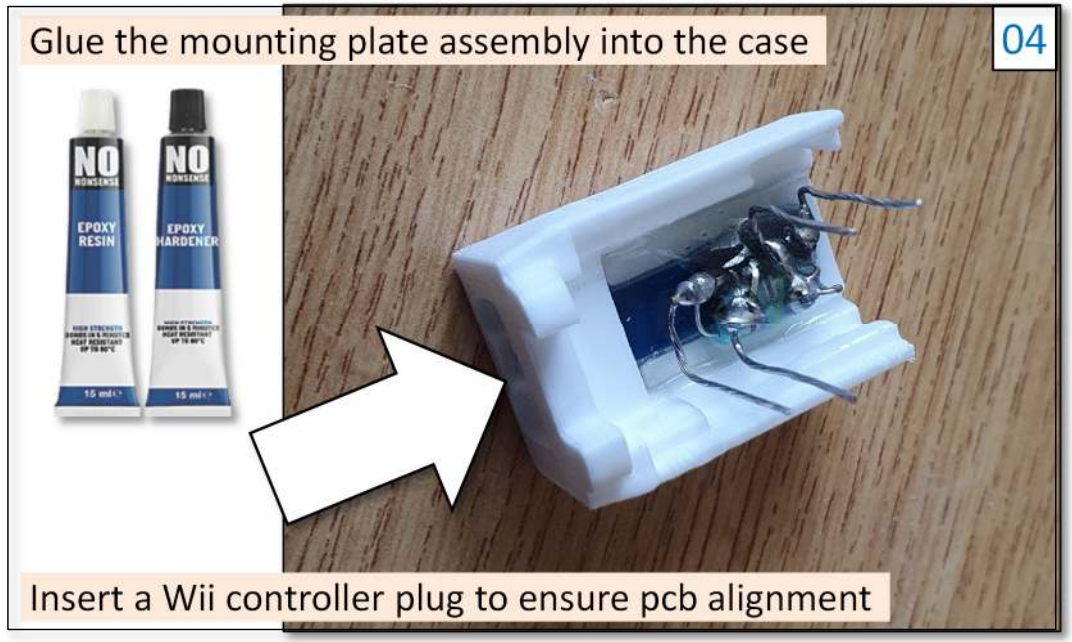
Allow generous wire lengths, and test for short circuits

03 Then attach the two 4k7 pullup resistors and 4 wires



Allow generous wire lengths, and test for short circuits

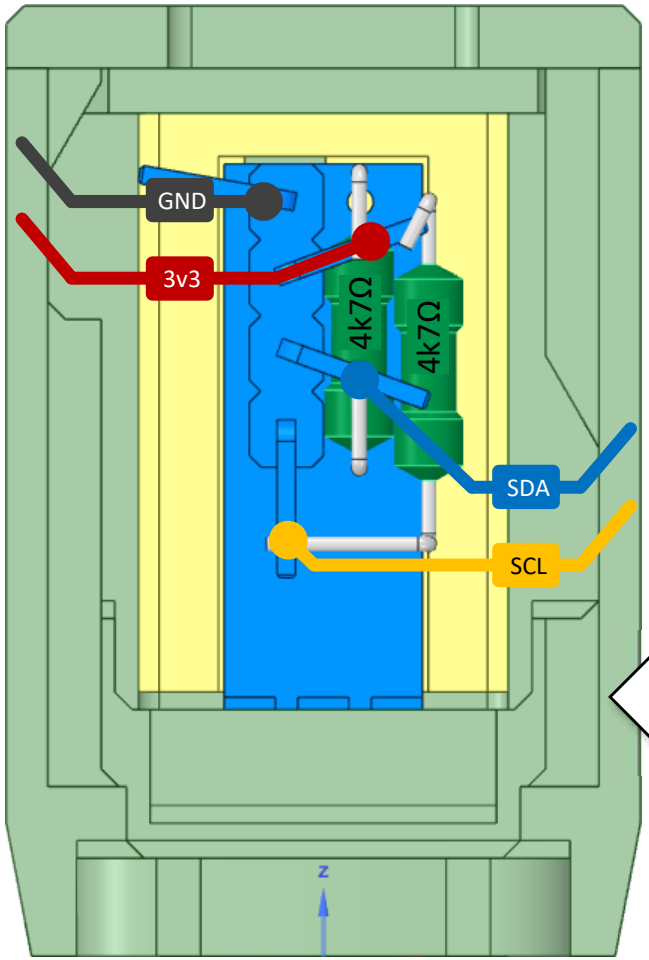
Glue the mounting plate assembly into the case 04

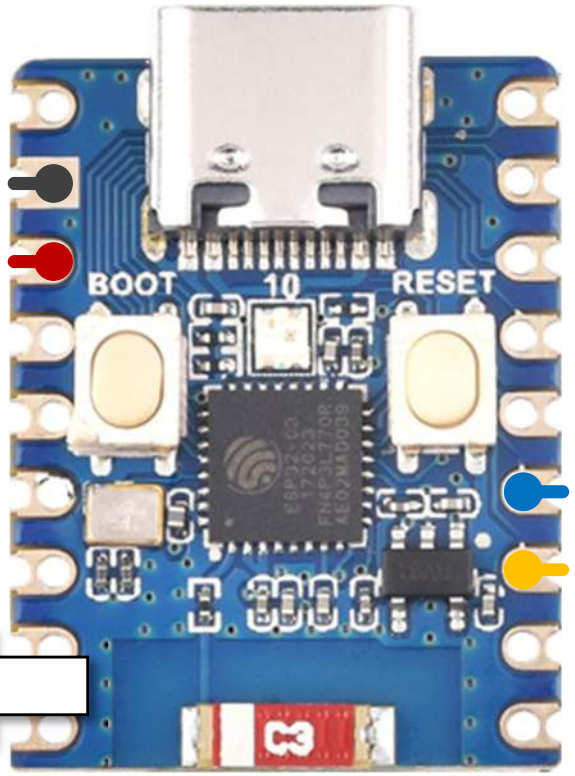


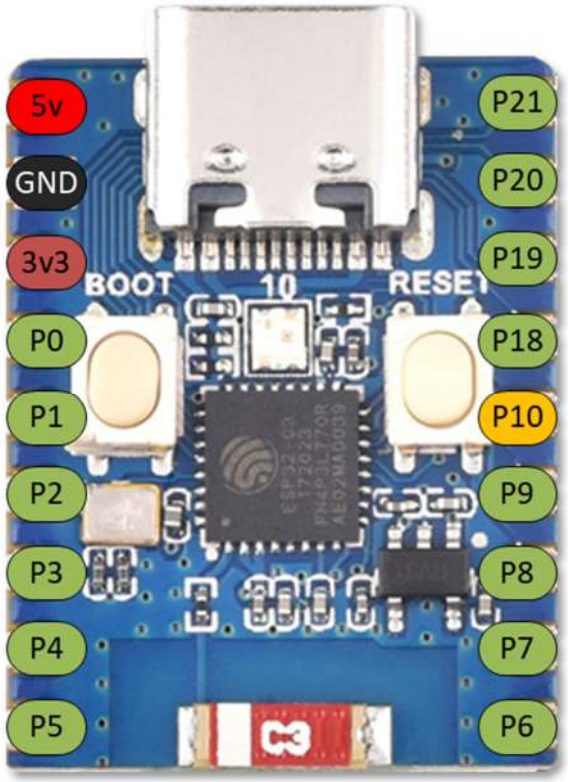
Insert a Wii controller plug to ensure pcb alignment

Case Wiring

Bend the four pins on the Wii adapter pcb as shown. Then glue the Wii adapter pcb into the sliding mounting plate. Then attach the two 4k7 resistors, between the 3v3 pin and their respective SDA and SCL pints. Additional wires are then brought outwards and upwards so that they feed through the respective hole in the ESP32-C3 pcb. Glue the slide into position with a Wii connector attached, to ensure alignment of the plate. Solder the four ESP32-C3 connections, then glue it into the case from the underside. If you use bare wires, like I did, take care to ensure that they do not touch and cause short circuits.



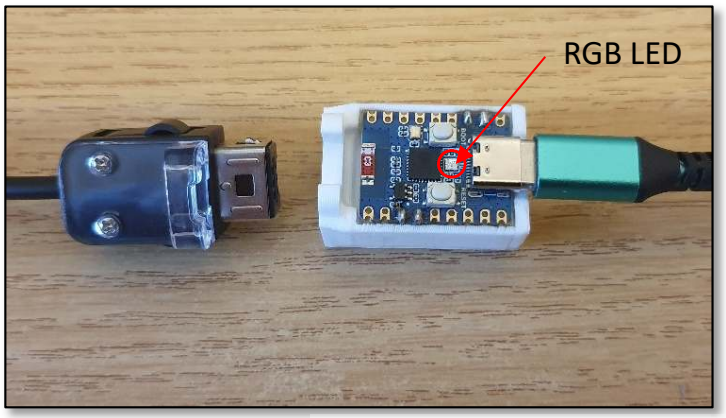




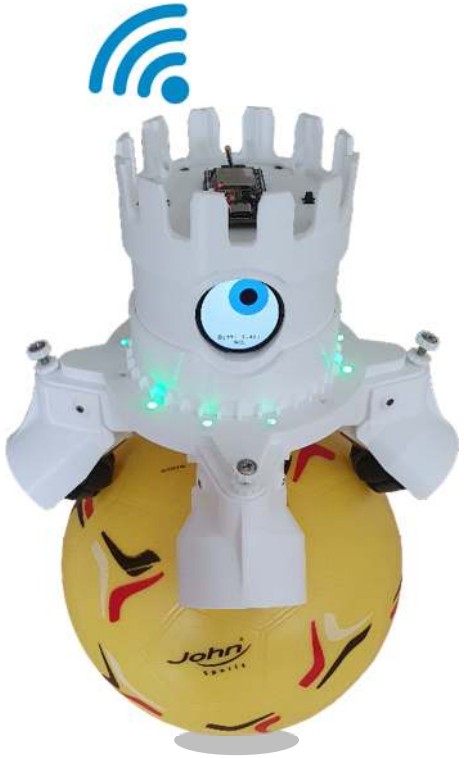
ESP32-C3-Zero pinout

Connection States

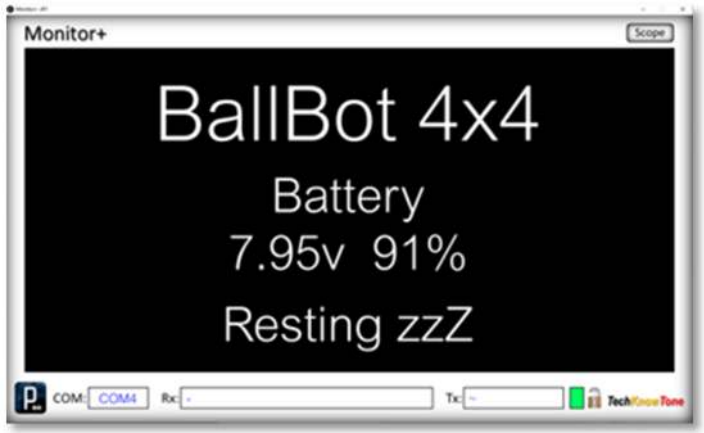
In this project we use the built-in RGB LED to provide visual status information.
Note, there is a small delay in powering up the ESP32-C3 due to its need to initialise its internal USB circuitry.



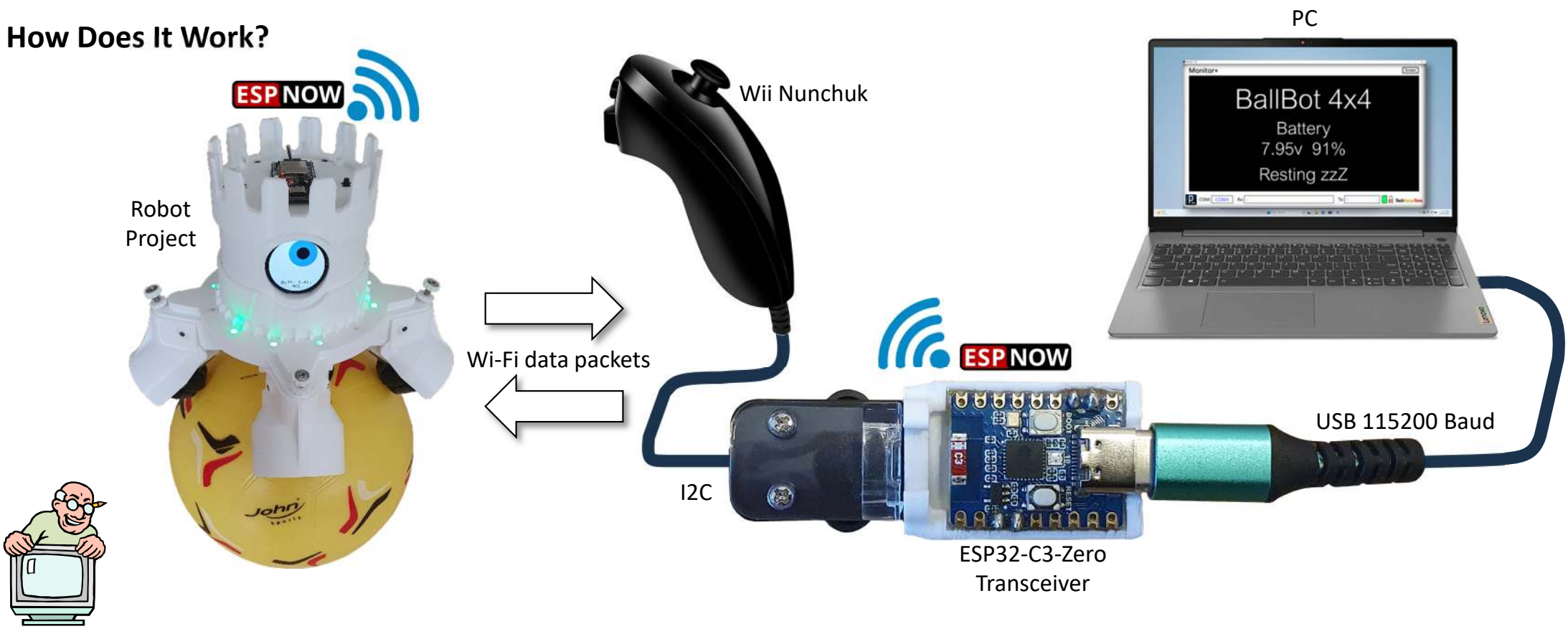
- When the ESP32-C3 is simply attached to a powered USB lead, the built-in RGB LED will flash on/off in the colour **RED**.
- When the ESP32-C3 is then attached to a Wii controller, the built-in RGB LED will flash on/off in the colour **PURPLE**.
- When the ESP32-C3 is then connected to a project, over Wi-Fi using ESP-NOW, but there is not Wii controller detected, the built-in RGB LED will flash on/off in the colour **GREEN**.
- When the ESP32-C3 is then connected to a project, over Wi-Fi using ESP-NOW, the built-in RGB LED will flash on/off in the colour **BLUE**. This is the condition you want to see in practise.



Once you have your project connected over Wi-Fi you can use the Monitor+ app to display interactive screens of data, produced by your project.



How Does It Work?



The ESP32-C3-Zero transceiver connects to a PC via USB, as a serial device, at 115200 Baud, and takes 5v power from the USB port. It also detects and connects to a Wii controller device, over the I2C bus; detecting the device type (Nunchuk, Classic or Classic Pro) and reading the controllers six registers every 40ms. And it connects to a project, over Wi-Fi using the ESP-NOW protocol. This protocol acts as a bi-directional interface, over which packets of data can be transferred asynchronously, in both directions.

Once the link is established, all data sent from the project, and received by the transceiver, is sent seamlessly to the serial port, and onto an app running on the PC, like Monitor+, or even the Arduino IDE serial monitor. In the initial handshake, the robot will declare its name, which could be significant for some applications.

Serial data received from the PC, by the transceiver, is then combined into packets along with data read from the Wii controller I2C registers, and sent to the project via ESP-NOW. The underlying packet structure therefore allows for a) a short packet, with only data from the Wii controller, no PC data available, b) a split packet, containing both Wii controller data, and serial data from the PC, or c) a packet only containing serial data from the PC, as the I2C interface is only read periodically. So data packets will vary in content, and in length (bytes) transferred.