**CO2 Transmitter with Raspberry Pi Pico**

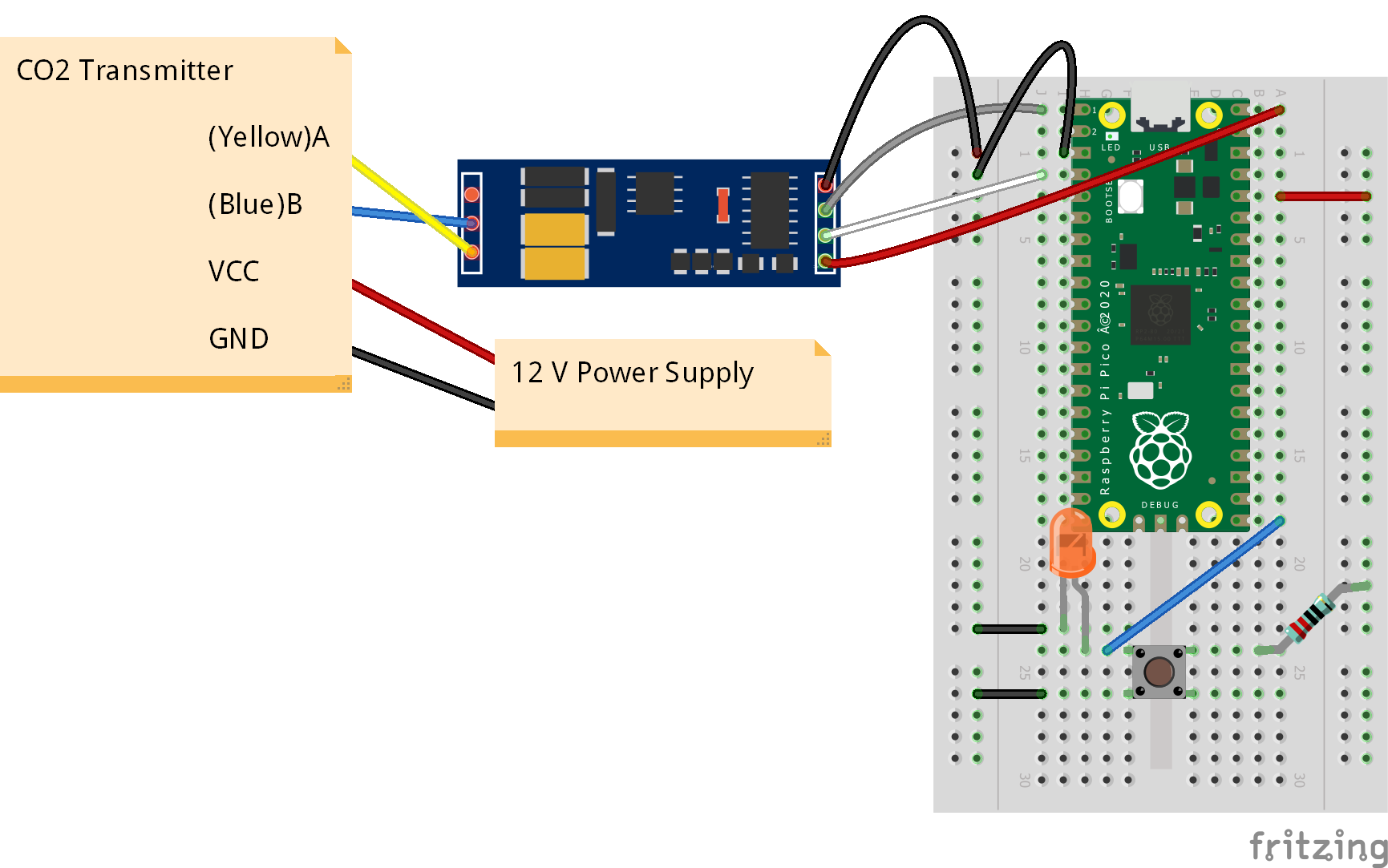
**Requirement**

* CO2 Transmitter Sensor
* Raspberry Pi Pico
* RS485 Auto Direction Module
* 12 V DC Adapter

**Language:** Micropython

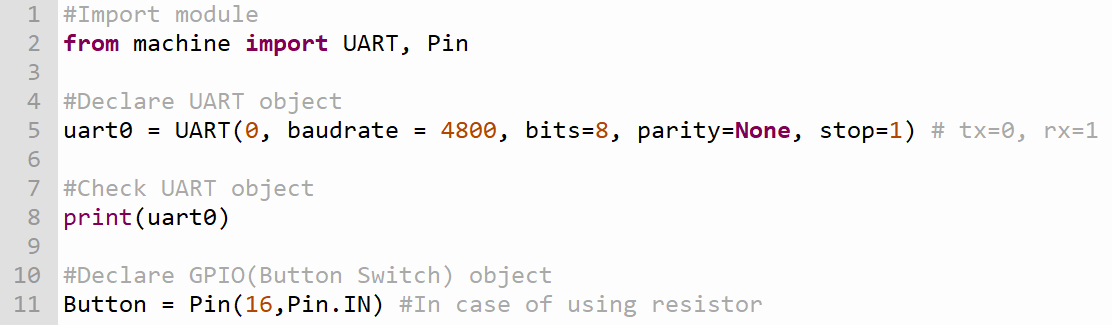
**IDE:** Thonny

**Wiring Diagram**

****

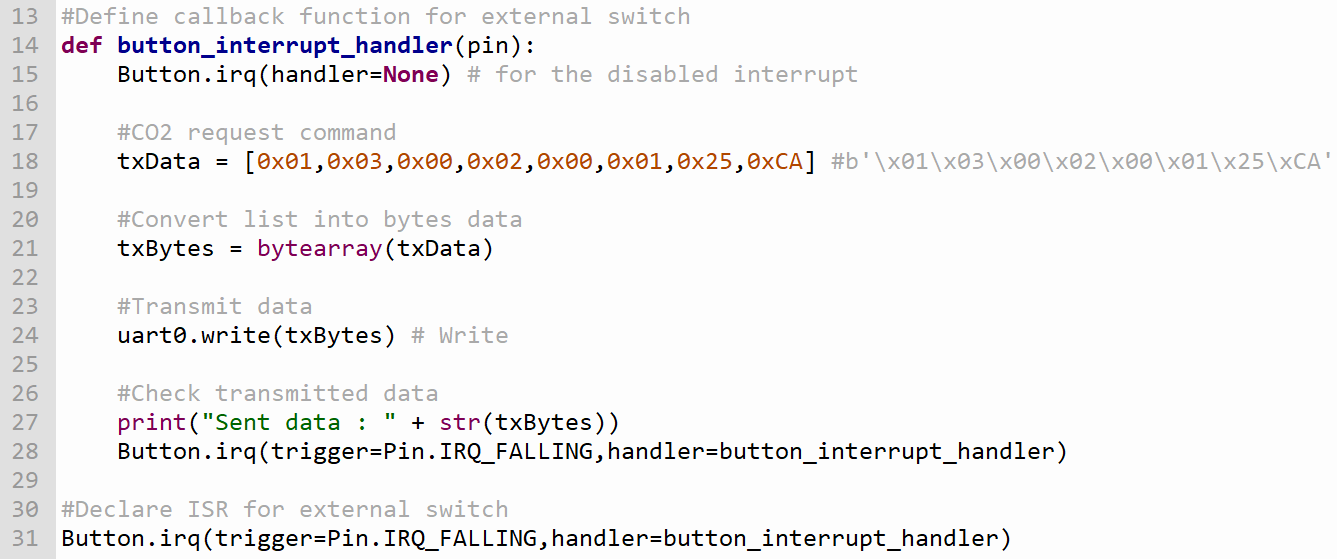
**Coding**

1. Import the module that we use in this program. Luckily, we only use “machine” as a module. Import “UART” and “Pin” for using UART and GPIO of our Pico. The declaration of pin is already show in the picture below.



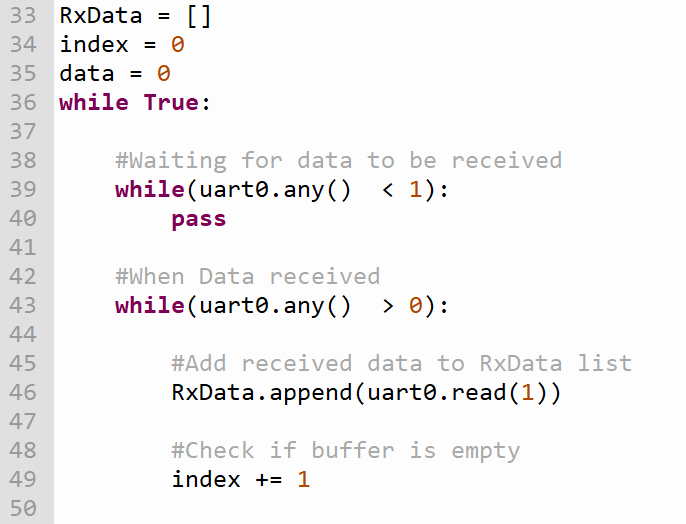
(Pic.1: Importing module and pin declaration)

1. Our propose is when we push the button, Pico will get the CO2 from CO2 transmitter. So, we use the push button switch as a trigger switch. To prevent button bouncing, we’ll use the switch as an interrupt by define the callback function (function that will be triggered when the switch is active) and declare the Interrupt Service Routine (ISR) to pin that switch used. As you can see, the push button switch that we wiring is active-low. So, assign the interrupt to trigger at falling edge. In order to get the data from CO2 transmitter, we have to send the hex command to it. So, we start declare the constant list as the command and then send it with the function in the picture below. We print the command in case of checking.



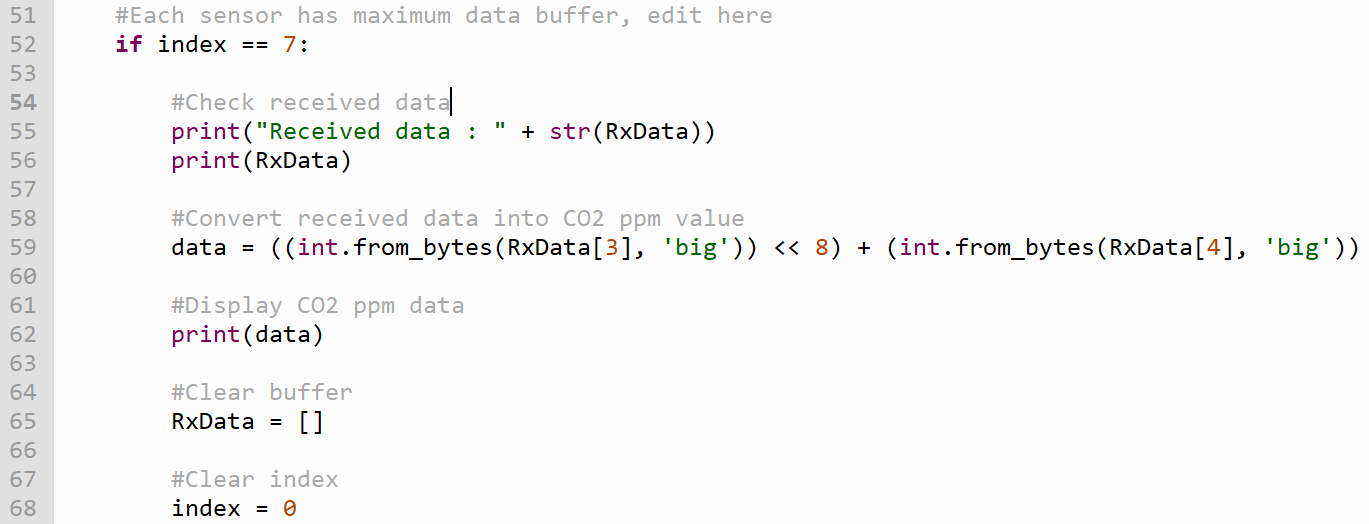
(Pic.2: Callback function define and data sending)

1. Let’s declare some variable. In this manual we will name it as “RxData”, “index” and “data”. “RxData” is the empty list that we use to store the buffer that sensor has sent. “index” is the counting variable that use to check and limit data that Pico receive. “data” is the value of CO2 in ppm unit after calculate the data that we received. Don’t forget do infinity loop as the main loop (all statement will be running in this loop). In the picture below, we use loop (in line 39) to wait for data. If the data has been received, Program will be running to another loop (in line 43). This loop will add the data buffer into “RxData” and “index” will increase.



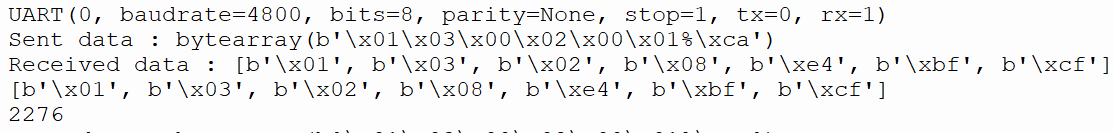
(Pic.3: Declaration of variables and inside the main loop)

1. In the datasheet of our sensor prove that the maximum data that will be received is 7 bytes. To prevent unwanted data, we will use “index” to count the number of bytes we received. If it is 7, we will stop received and calculate the value of CO2. In calculation, we use only the data in 3rd and 4th byte that we receive. So, we need to shift the value 3rd byte before add the value of 4th byte. Unfortunately, that the data that we receive is “byte” object and it can’t be shift. Before we shift the value, we need to convert the data type into “int” so we can use bitwise operation. The instruction is shown in the picture below.

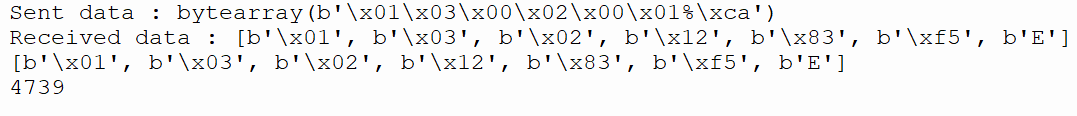


(Pic.4: CO2 Calculation)

1. For the result, the value of CO2 has been stored in the “data”.



(Pic.5: The result of program when we start.)



(Pic.5: The result of program when we breath near the sensor.)

