

I mean all this because I have seen several examples of hobbyists gutting appliances and modifying them completely, or sometimes just to “recover” components, as if they were nutrients for uncertain future projects.

It is preferable that they rest in peace (the artefacts, of course) complete, well kept if we have a storage place, until the day when, for example, a child asks for one of them.

Dad, what is that? - A phone, son. What you see is a phone and if you put your forefinger there you will see that you can turn it and it will dial a number.- And does it work Dad, can we use it?

Yes, of course ...erm, I think we would have to stirp the wire because we don't have a plug like that at home, but ...well, we better leave it for another day.

This conversation is absolutely real; and

the device that worked in my great-grandmother's house, made in Sweden a long time ago, is still waiting for me to one day find or conform something that works for it, because we really don't want to alter its plug.

If you are in panne in the middle of the Sahara desert and one of these personages appears to you, surely he will ask you to please draw a lamb for him. The request is not like a court order, but it is just as irrevocable.

Dad, I remember that I saw that you had a joystick stored in a box. May I take a look at it?

Zas! He is already 7 years old (the child, the joystick more than 20) and he is very interested in everything that may attract his attention, which by the way is quite a lot. And he has also just discovered that there are very realistic flight simulation

programs and that no matter how skilled you are with the keyboard...

May I... May I... May I take a look at it?

MAY DAY!!!

Inside the box actually rests the joystick, impeccable, with its very brief instructions manual, and its purchase receipt dated December 2000.

Aha! **DB-15**, in other words, a plug to connect to the game port of the sound card of the old computer, with 15 pins!

Arduino will surely give us a hand; we will see.

Second question

(Since the first was “May I take a look at it?” repeated several times)

Can you connect this device or similar through an Arduino board and get any results?

Answer: Yes.

The sense of including this question and its answer here is for those who have seen that it works, for example the KY-023 joystick module in Arduino UNO with which you can turn on LEDs arranged in a cross or also act on servo motors, but it wouldn't behave like a joystick connected to a computer as expected.

I should also mention that it was the first thing we did. I went to our box of 37 sensors to check if there were one of the famous KY-023 to do the first tests as you can see in Figure 2, before moving on with the big one. Because at this point I had already inferred that there should be several lines of code to read and some libraries to download.

But I didn't know that we should acquire something new for our beloved and modestly supplied collection of electronic components.

So, third question.

Could the joystick (with its 15 pins plug) through the Arduino board communicate with the flight simulation program and take advantage of all its features (buttons, rudder, throttle)?

Answer: Yes but no but yes.

Yes, but not if you use the usual boards such as UNO, Nano or Mini based on the ATmega 328 chip (*). And the answer would be **yes** if you choose the **Pro Micro** board equipped with the **ATmega 32U4** chip (**), and in this particular case We find and recommend the variant that instead of the micro usb connector, it brings a **mini usb**, more manageable and resistant. (Although we actually get it for

less than **u\$s 5**, remember that users may be under 8 years old!)

(*) There is a way to do this , it consist in reprogramming the usb protocols translator chip ATmega16U2 -which is the other or second chip- in the UNO board, but it surpasses this work (also mi knowledge), although at first glance we realize that we would have half of resources. By the way the ATmega32U4 chip is very powerful and makes the Pro Micro board a nice masterpiece.

(**)The Arduino Leonardo and Micro are also microcontroller boards based on the *ATmega32U4*, and of course they would work the same way.

Ok so what do we need besides the Pro Micro?

Solder. You don't have to be an expert by any means, but the Pro Micro board (unless we already have it) can come with the pins without soldering. And it is also necessary to solder resistors, some wires, and connector terminals to each other.

One DB15 Female D-Sub Solder Type

Connector

Four 100K Ω resistors

Six Dupont Jumper Wires Male to Male 20 cm Length (which will be cut in half to connect its terminals to the breadboard and solder the opposite ends to the DB-15 connector), preferably of different colors, because they are soldered and with the connector clamped in place, it is difficult to identify them and connect the corresponding pin.

One 170-point mini breadboard (usually comes with double-side tape).

Pliers, tweezers, wire strippers, soldernig gadgets, magnifying glass, enough light, etc.

Any piece or assembly of parts that allows something similar to what we achive with pieces of a Meccano game,

and whose final weight of 75 grams is enough to remain stable even if the cables move.

There is almost always more than one way of doing things. We opted to develop a detachable device. In a matter of hours we can find something, like for example an RC plane, and we would have to recover our Pro Micro board if it was useful for the new project.

Although my children are delighted to "lend" me their toys, including Arduinos, so that we can do new experiments, we all know that they must return to their original state immediately at the slightest requirement. They know exactly what each piece of Lego is and what place it occupies in what can be a clock hand, or a plastic separator between shields and supports. But it is not a prodigy of memory, but a relentless power of control!

Of course, if you are not harassed by a couple of Sheriffs of Nottingham, or you are looking for something smaller and more permanent, soldering everything to an experimental board, without having to have printed circuit, may be the best solution. Among the pages I have read as research for this project, I found mention of distortions in the signals received by the PC due to connection failures to the breadboard, so it is recommended to solder. So far in our tests everything works perfectly as we have arranged.

This joystick consists of four

potentiometers, two that respond to the X and Y axes, another for the throttle and another for the rudder, with variable resistance values that could be between zero and 100 k Ω approx. In each of them, one end of the potentiometer is connected to the 5 volt input pin and the center contact is connected to the analog input of the joystick. The other contact of the potentiometer is left unconnected. This arrangement requires ground discharge through four 100k Ω resistors, one for each analog input from the four joystick potentiometers, as we will see in the schematic circuit. The digital pins correspond to the operation of the buttons.

To use the Pro Micro board, you need to download and install a driver.

Here is the link for downloading and installation instructions for both Linux, Mac or Windows:

<https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide#installing-windows>

And now the code for the Arduino IDE

```
// To make use of this Joystic Library, we  
must first download it at the address:
```

```
// https://github.com/MHeironimus/  
ArduinoJoystickLibrary
```

```
#include <Joystick.h>
```

```
#define BUTTON_1 4
```

```
#define BUTTON_2 5
```

```
#define BUTTON_3 6
```

```
#define BUTTON_4 7
```

```
#define DELAY 100
```

```
Joystick_ Joystick;
```

```
void setup() {
```

```
pinMode(BUTTON_1, INPUT_PULLUP);
pinMode(BUTTON_2, INPUT_PULLUP);
pinMode(BUTTON_3, INPUT_PULLUP);
pinMode(BUTTON_4, INPUT_PULLUP);
Joystick.begin();
Joystick.setYAxisRange(1023, 0); // You
can reverse the order of these values to
shift up or down
Joystick.setXAxisRange(1023, 0);
Joystick.setThrottleRange(1023, 0);
Joystick.setRudderRange(1023, 0);
}
```

```
void loop() {
  Joystick.setXAxis(analogRead(A0));
  Joystick.setYAxis(analogRead(A1));
  Joystick.setThrottle(analogRead(A2));
  Joystick.setRudder(analogRead(A3));

  if (digitalRead(BUTTON_1) == LOW) {
    Joystick.setButton(0, 1);
  }
  else{
    Joystick.setButton(0, 0);
  }
}
```

```
}
```

```
if (digitalRead(BUTTON_2) == LOW) {  
  Joystick.setButton(1, 1);
```

```
}
```

```
else{
```

```
  Joystick.setButton(1, 0);
```

```
}
```

```
if (digitalRead(BUTTON_3) == LOW) {  
  Joystick.setButton(2, 1);
```

```
}
```

```
else{
```

```
  Joystick.setButton(2, 0);
```

```
}
```

```
if (digitalRead(BUTTON_4) == LOW) {  
  Joystick.setButton(3, 1);
```

```
}
```

```
else{
```

```
  Joystick.setButton(3, 0);
```

```
}
```

```
delay(DELAY);
```

}

From now on you can fly alone, it's part of
the game.