Arduino and its shields are a success story. Why? Because it is a system with modular and standardized components. This made the entry into microcontrollers easy for a whole generation of electronic newbies.

Now, we are at the verge of new concepts, which want to drive the standardization to a new level and promises to make things even easier than before. The promised land of marketing is to build IOT devices for everybody. This attracted my attention. Today, I want to look inside one of these new systems: The cowtails from Elecrow.

Let’s start.

Lastly, Massimo Banzi from Arduino presented a Kickstarter initiative, called ESLOV, with the goal of a new system: Hardware and software should become easy like Lego. Microduino presented two different systems which also tend into this direction. And Wemos tries to create some sort of shields for the ESP8266. These will reviewed later.

Elecrow uses a similar hardware approach than ESLOW. They connect their modules with short wires. This is, why they call it “cowtail”. And because, here in Switzerland, we have plenty of cows including tails, this system cached my attention.

As said, the system consists of small modules, connection wires between them, a manual with many different experiments, and they provide also the example sketches which are necessary for the experiments. You can buy the individual modules, or you can start with a kit. I ordered both, a kit, and some additional modules. The kit came in a nice box. So, lets unpackage everything and see, what we have got.

Each device has its own little bag, and even a silica gel is included to keep humidity down. There are many sensors and actuators included in this kit. You can check them on the homepage of Elecrow. A link is provided in the comment. At the end, we get to the two important pieces: An Arduino and a special shield.

Already now it becomes clear, that the system has a star architecture: One central hub has to be connected to the different devices. The devices only have one connection, you cannot “daisy-chain them.

If we have a closer look at the base shield, we see a lot of connectors, some have three pins, and some have 4 pins. And they are named according what we well know from Arduino:

D port– for Digital

A port – for analog

U port – for UART, or Serial

I port – for I2C bus

So, here they are joust breakouts of the Arduino UNO standard pins. We have just more pins, because each connector has its ground and Vcc pin. Apparently, all is 5-volt logic.

So, according this logic, not all devices can be connected to each connector, even if the connector might fit. According our experience with Arduinos, connecting a sensor at the wrong port, usually will not destroy a part, it just will not work.

To help beginners, we find on each device the letter of the port it has to be connected. In my kit, there are many parts with D, but none with an A. Fortunately, I ordered also one, which carries an A. It is a strain gauge to measure weight.

The other device, which I thought, should carry an A is the joystick. But it carries an I. First I thought, that they included a I2C controller on the device itself, but then, I discovered, that they just use a trick, which works on some Arduinos: Because the I2C pins are A4 and A5, they just use them as analog pins. So, if you use the joystick, you cannot use an I2C device and vice versa. On one hand, I would wish, they really would have included an I2C driver. But, then, of course, they would get other problems with existing libraries etc. So, maybe, the compromise is ok.

But we see here the advantages and disadvantages of the chosen architecture. It is very close to the standard Arduino architecture, with all the hardware dependencies. The advantage clearly is, that we can use the complete Arduino community stuff, and everything we learn here, can be used if we later on go to the “real stuff”.

But let’s continue. In the meantime, I connected all tails to the “cows”. These connectors are really a big, big advantage over the Dupont wires. You do not need to look, where ground or VCC is, you cannot confuse the two and destroy a part, and they do not become lose. Great, also for a veteran like me! I have to confess, I never liked the single Dupont wires. This is, why I usually use empty shells as presented in my video #12. But this is definitely more standardization, because all devices also use the same pin order.

So, let’s now do a first test: Let’s try a touch sensor. Searching in the downloaded manual for the word “touch” brings me to lesson 1. The goal is, to control LEDs with a touch sensor.

I follow the instructions and, again, enjoy the comfort of the nice connectors. Because the tails are still “ring-tailed” more like pig-tails than cowtails, I get my third hand of video #77. After connecting the Arduino to the USB, the LEDs start to blink. So, the guys from Elecrow have already loaded a sketch. Maybe, this was written somewhere in the manual, but, as engineers, we only read the manuals if we cannot avoid it. And this is usually, if something went wrong already.

But, the loaded sketch does not fit lesson 1. So, I try to download the “Cowtail Advamced Kit Demo code” from the provided link in the manual, but, unfortunately, I only get a 404 error. But, with some searching, I found the demo code. I will post the link in the description and also send a feedback to Elecrow.

During my searching I discovered this page: They have much more “cows” with a tail than I thought.

Coming back to the example code. After uploading, it works. Success!

Now, I want to test a more complex experiment: A Temperature and humidity sensor with display. This is lesson 7. I start with the same steps as before: Load the sketch, connect the “cowtails according the drawing in the manual, and start. The IDE reports an error, because this sketch needs two libraries. I have to copy the provided libraries into my libraries folder. Again, it would have been written in the manual. Did you know the important abbreviation: RTFM, read the fucking manual?…

After the libraries were copied, everything works like a charm. And I have my humidity and temperature meter in my lab. With the ESP8266 cowtail provided with the kit I could even send this data to the cloud. This ESP8266 is a cowtail, and it is configured in the AT mode. But this exceeds definitively the “beginners” manual, which ends with a “hello world” example with the ESP8266.

Apropos ESP8266. Of course, I ordered also another ESP8266 board, this time as a “host” system. This board replaces the Arduino UNO and the cowtail-shield. Because of the limited I/O of the ESP, this board is smaller and has much less connectors. Still, it has an I2C, two digital, and one analog connector. In addition, it also has a UART.

I wanted to try the temperature and humidity example with the ESP. But the Liquid Cristal library did not compile. Then, I looked for the proper documentation and found also a manual for the ESP host. It has his own lessons, but based on the IOT kit’s cowtails, which are different from mine. So, I tried the motion detector, and it works. The manual provides a link to code for both, the LUA and the Arduino environment. Unfortunately, the manual and the code contains small errors. I think, it has to be quality checked. I did not try the LUA code.

At the end, we can put all together in the box provided with the kit and store it away till the next use.

Where does this new kit fit into the “new” Arduino world?

In my opinion, it is more a combination of an Arduino and a sensor kit or individual sensors. The price of 55$ is, compared with the individual parts, is not much more expensive. However, it adds quite some value. At least, the Arduino kit. As already said, the connectors are a real comfort, and for beginners, a big help to avoid defective parts. The manual has a selection of nice lessons. The quality of the Arduino manual is good, and the code worked. They just have to fix the “missing link. Then, this is a very good Christmas gift for a person, who wants to start with Arduino.

The selection of different cowtails is impressive. And, even, if you miss one, you could probably build your own one.

The ESP8266 board, however, is a different story. The ESP chip itself has already only a few I/O pins. But this board is too minimalistic. Maybe, if you are interested in LUA only, this might be a solution. But if you work with the Arduino IDE, go for the Arduino kit. You get much more for your money. And even, if the manual and the code for the ESP8266 is fixed, still the problem of voltage level conversion between 3.3 and 5 volt is not solved. At least, I did not see any level shifters on the board.

How does it compare with the ESLOV Kickstarter? This is still a conventional approach where you have to learn some hardware and software and you can fail if you want to do your own stuff. ESLOV wants to make programming much easier. I wonder, what value you get out of such a simplification. As an engineer, I think, it is still necessary to get your hands dirty if you want to play with microcontrollers. And these cowtails make it a little easier and more comfortable.

I hope, this video was useful or at least interesting for you. Bye.