You all know, that I like the Wemos Mini because it is ideal for many projects: Not much bigger than an ESP-12E module, but everything built in that you can start right away. You can solder wires to it, and you can upload sketches and monitor with the Serial connection.

Recently, I discovered, that Wemos and others added a new twist to it: They created shields for the Wemos Mini. Today, I will play around with them and you get also my opinion about them.

In video #89 I looked at the Crowtails from Elecrow, which also try something similar: A modular, and easy-to use concept. I found it useful for beginners. But I did not like the ESP8266 board, because it had only a few pins to connect to it.

The Wemos shields are different. As the Arduino shields, they have the same form factor as the Wemos mini and also exactly the same pin layout. On Aliexpress and on Banggood, currently, you get about 12 different shields:

A relay shield, a motor shield, a battery shield, an OLED display shield, three different temperature sensor shields, a prototype shield, a double socket shield, an SD card shield, a WS2812 shield, and a BMP180 shield.

All shields come with three different header pins: A female, a male, and a female with long legs which can be used as a male on the other side. You have to choose and solder them by yourself. We will see later, that this is a good concept.

All shields work with either the Wemos Mini or the mini pro. The difference between the mini and the mini pro is the flash and the FCC certification. The additional flash cannot be used in the Arduino environment, at least not without tricks. So, I see no advantage for the mini pro. At least, it can be programmed like a NodeMCU 1.0 version.

One of the most important things with shields is the pin assignment. Especially with the ESP8266, we only have a few pins, and also these cannot be used for all purposes. So, I checked all shields and found this chart: The Oled, the BMP10, and the Motor shield all use I2C bus on pin D2 and D3. This is a very good concept, because devices can be run in parallel on this bus, so you can use two or more shields in parallel. The Motor and the OLED shields have also selectable addresses, so you can use more than one of the same shield in one setup. The BMP180 has no selectable address. But it would anyway not make sense to have two of these sensors in close proximity, because temperature and pressure anyway is the same…

The DHT shields use a so called “one Wire” interface. Its pin is on D4 and you should be able to connect many of them in parallel. However, I did not test it. Maybe a viewer has experience with these devices.

The SD card uses completely different pins, so I do not expect problems. This is different with the relay. It uses pin D1. use So, the relay is not usable with any I2C device, which is a pitty. And the DS18B20 and the Neopixel both use pin D2, which limit their usage, too. I would have preferred to have a switchable pin for all these devices (even with solder bridges). Maybe they will do this in a later release because it would not add any manufacturing cost but the maker could choose according its design. Here, we clearly see the limitations of these “shield designs”.

At least, you have now a “compatibility matrix if you plan a device.

Enough ranted. Let’s continue with the work. We have to solder the pin headers to the boards. Here you have to pay attention, because otherwise, you run into problems and might not be able to use your shields. As many things, these boards have two sides, and if you chose the wrong side, the shield is unusable, because it is hard to unsolder header pins. I started with the dual base and used it as a reference. I placed it with the text on top. This makes sure, that all other shields are “sunny side” up. If you start wrong, you might discover later, that the OLED can only mounted “sunny side down”. I decided, that the base should be the base and therefore, soldered only female pins with short legs to it. Now, you know where the reset pin is. I used this one for easy reference.

When you start to solder the next shield, just place it above the base plate for reference and insert the headers accordingly.

But which header is the best? Generally, I choose the long female headers for all shields, exept the ones where it makes no sense that you place a shield above them. Which clearly is the OLED, Here, I only soldered male headers at the bottom of the shield. This should apply also to the Neopixel, but I made a mistake. And the same applies for the proto shield, where I discovered later, that I want to use it for a button. A tip for the proto shield: solder the headers last. It makes your live easier…

So, we have now the shields prepared. Now, we want to get something running.

Fortunately, Wemos provides example files for their boards. You find the link in the description. So, I wanted to build the “clock” example, because I think, this is very neat with this small OLED. So, I downloaded the zip file and installed it in my “libraries” folder. After restart I found the examples where expected and wanted to compile the clock. Unfortunately, I got error messages. All examples base on libraries from either Sparkfund or adafruit and you have to install these libraries if you did not do it before. Just google with the name of the “include” file and you find the repositories.

After that, the sketch compiled and also run without any problem. But it did not show the actual time. As a Swiss, this is not at all acceptable for me. So, I enhanced the sketch with the internet time of video #71. Now it is precise like an atomic clock for the next few thousand years. This is, how we solve issues here…

Of course, people from all other nations can download my sketch from github…

Now I wanted to test a sensor. The easiest is the DHT pro. So, the same procedure: Stack the shield wherever you want, select the example file, install the libraries, compile and upload. Everything works. But of course, only in Serial monitor. Not acceptable, again. So, same procedure: Enhance the sketch with the OLED functionality and stack the OLED shield ion top. Also here, you find the sketch on github.

Very neat. It is fun to play with these small devices, even for an old guy like me. And the result looks really nice, it would also fit in a small box. But, of course, I want more: I want to have this device battery powered. Nothing simpler than that. I use the battery shield. It exposes a 2.54mm JST connector for a single cell LiPo battery. I had such a battery, but with a wrong connector. Fortunately, with the help of my connector assortment from my last mailbag video, and the crimper from video #39, I was able to build the missing link. Now, the whole thing runs from battery. The shield has a buck converter to 5V and a LiPo loading circuit. So, you just connect a USB cable to load the battery. A great shield also for other projects.

But frequent viewers know, that the video cannot end like that: I want to be able to change the sketch on the fly from IOTappstory.com. So, I transplanted the two sketches into a suitable format. If you do not know this concept, please watch video #105. Most of you know, that I had to add a button to the Wemos to trigger uploads and configuration. So, I used the protoshield for that purpose and connected a push button with a capacitor and a pullup resistor between D3 and GND. And recently, I found very neat push buttons with a built-in LED. So, I used this LED as an additional indicator on pin…. But wait: What did I say before: that the pin should be selectable. So, I added a simple dupont connector to one pin. Now, I can select the pin.

Now, the whole story is ready to rumble: I create my new Wemos device in IOTappstory.com. This time, I call it Wemos red because of the red point. Then, I create a project with my device and the Clock sketch. Now, I compile and upload the iotappstoryloader file from Github, connect my Iphone to the Wemos board to enter the credentials, and watch what happens. Here you see it: The Chinese-Swiss precision clock appears!

Now, I go back to IOTappStory.com and change the app to the temperature meter, press the button for about three seconds, and the shields are now a temperature meter.

Summarized, we have here a concept which is similar to others. Its flexibility is quite small, because of the few pins of the ESP. But I like it, because it is very small and you can build real things with it. They are so small, that you can just put it in a box. And with the proto shield, you can extend its reach. I already ordered 10 more… Many of my future ESP projects will contain a Wemos and a proto shield as a minimum configuration. So, I do not have to solder the wires directly to the Wemos board and small components like a transistor or a pullup fit neatly on this board.

If I do not need it anymore, I still can reuse the Wemos board with ESP. And maybe some other prefabricated shields will appear over time. Hopefully, with variable pin assignment…

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