



HackerBox 0062: Watts Up



by HackerBoxes

Greetings to HackerBox Hackers around the world!

With HackerBox 0062, you will discover two new embedded processing systems: The Bouffalo Labs BL602 RISC-V SoC and the new S2 variant of the popular Espressif ESP32. Both feature low power operations as well as wireless communication functionality. Also, explore USB power delivery and switch-mode boost power supplies for big results from small batteries. Additional topics include running Linux on Windows, integrating temperature and pressure sensors, and the relationships between current, work, power, and battery capacity.

HackerBoxes is the monthly subscription box service for enthusiasts of electronics and computer technology - Hardware Hackers - The Dreamers of Dreams.

There is a wealth of information for current and prospective members in the [HackerBoxes FAQ](#). Almost all of the non-technical support emails that we receive are already answered there, so we really appreciate your taking a few minutes to read the FAQ.

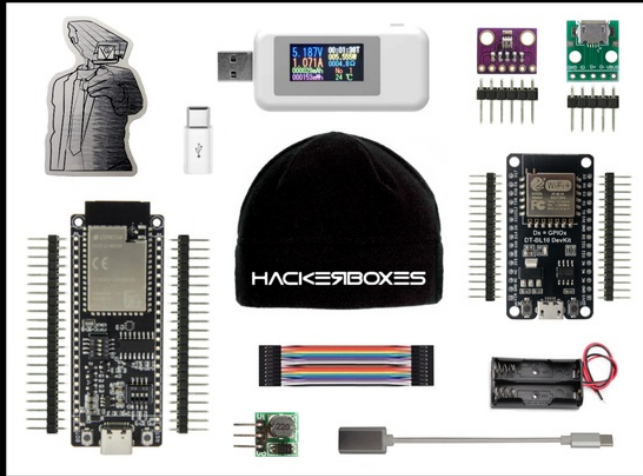
Supplies:

This Instructable contains information for getting started with HackerBox 0062. The full box contents are listed on the [product page for HackerBox 0062](#) where the box is also available for purchase while supplies last. If you would like to automatically receive a HackerBox like this right in your mailbox each month with a \$15 discount, you can subscribe at [HackerBoxes.com](#) and join the revolution!

A soldering iron, solder, and basic soldering tools are generally needed to work on the monthly HackerBox. A computer for running software tools is also required. Have a look at the HackerBox Deluxe Starter Workshop for a set of basic tools and a wide array of introductory activities and experiments.

Most importantly, you will need a sense of adventure, hacker spirit, patience, and curiosity. Building and experimenting with electronics, while very rewarding, can be tricky, challenging, and even frustrating at times. The goal is progress, not perfection. When you persist and enjoy the adventure, a great deal of satisfaction can be derived from this hobby. Take each step slowly, mind the details, and don't be afraid to ask for help.

HACKERBOX 0062



Step 1: Watts Up With USB Power

The USB Multimeter can measure current, voltage, power bank charging, battery capacity, and more. The meter can be placed inline between any USB power source and USB power sink to see how much current (in Amps) is being drawn into the sink. For example, try it out with a mobile charger for phone or tablet, with a microprocessor board, or any other USB device.

Generally, USB provides 5 volts DC. We can use a little USB break-out module to access the 5V pin and the GND pin and use them as a quick 5V power supply for an electronics project. This is sort of like a (nearly) perfect 5V battery.

You can connect a 100 Ohm resistor between the VBUS and GND pins and notice that the current on the meter goes up from 0A to: $I = V/R = 5V / 100 \text{ Ohms} = 0.05A$. The power supply is doing work against the resistor. Electrical work on a load usually moves something (like a motor), generates light, or generates heat. In this instance the resistive load turns all of the applied energy into heat since nothing is moving and no light is generated.

If we switch to a 50 Ohm resistor, the current drawn is: $I = 5V / 50 \text{ Ohms} = 0.1A$

The current went up when we lowered the resistance of the load. More water flows (current) when the pipe is less restricted (lower resistance).

Be careful not to draw too much current. Which means do not use too small of a resistor. In the extreme case, if we short 5V to ground with a wire, that is effectively connecting a zero ohm resistor. What happens to $I=V/R$ as R gets smaller and smaller? Current (I) gets bigger and bigger. At $R=0$, current (I) will try to go to infinity. However an infinite amount of water cannot flow, so we have a big problem. You might know from programming, never divide by zero, right? Current (I)

can actually provide various different voltage levels up to 20V. Learn more from this video about [Power Delivery and Why USB Type-C is Awesome](#).

Volts, Amps, Ohms, Coulombs, Watts, Joules... &\$%#@!

All these units can get confusing. The Ohm's Law Triangle and the Power Triangle shown here can be quite

tries to go to infinity and BANG - the smoke comes out! This is why hardware hackers do it without shorts.

Does USB always gives us 5V?

Traditionally, yes. Or at least close enough. Sometimes it is 4.8V or sometimes 5.1V. Nobody's perfect.

If you try to draw more current from a supply than it is rated to provide, the voltage will usually start to drop below the nominal rated output. Keep drawing more and more current and bad things will happen. The voltage dropping is sometimes called the elasticity of the supply. The opposite of elasticity is stiffness. A stiffer supply will hold a steady voltage output under increasingly challenging loads (e.g. smaller resistances, bigger motors, more LEDs). You have seen this if you try to run a Raspberry Pi (or a bunch of RGB LEDs) using a small (weaker) USB phone charger even though that charger seems to work fine for the lower demands of say an Arduino board with no added load.

It is helpful to understand that the current rating of a supply is generally a maximum amount at which the supply can maintain the rated voltage output. It is the amount of current that CAN be drawn out and not the amount of current that is FORCED OUT. So you can hook a circuit that uses 0.05A at 5V to a 3A power supply and it will only draw 0.05A. The 3A of current is merely what is available and (unlike voltage) is not what is actually forced into the load. When replacing a supply, you want at least as much current rating as you had before (or what the system uses) but it is not a problem to have a supply with a higher current rating. The current available to be supplied and the current drawn do not have to match. Of course, the voltages do have to match!


There is yet another twist with the assumption that USB provides 5V. As is often the case with electronics, technology is always changing. Newer versions of USB

useful. You might also want to watch these two videos:



Units of Work, Energy, and Power

Application to Current, Voltage, and Battery Capacity

If the cold winter air is drawing too many Watts from your head, you should put on a warm HackerBox fleece cap. The cap will serve as a resistor against the flow of thermal energy from your processing unit into the surrounding environment. You will feel warm, but look cool!



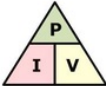

Ohm's Law Triangle



$$V = I \times R \quad I = \frac{V}{R} \quad R = \frac{V}{I}$$

V: Voltage in VOLTS
Difference in Electric Potential

I: Current in AMPS
Flow Rate of Electric Charge per Unit Time

Power Triangle



$$P = I \times V \quad I = \frac{P}{V} \quad V = \frac{P}{I}$$

R: Resistance in OHMS
Size of Load upon which Work is Done

P: Power in WATTS
Work Done upon Load per Unit Time

Step 2: ESP32-S2 Vs ESP32

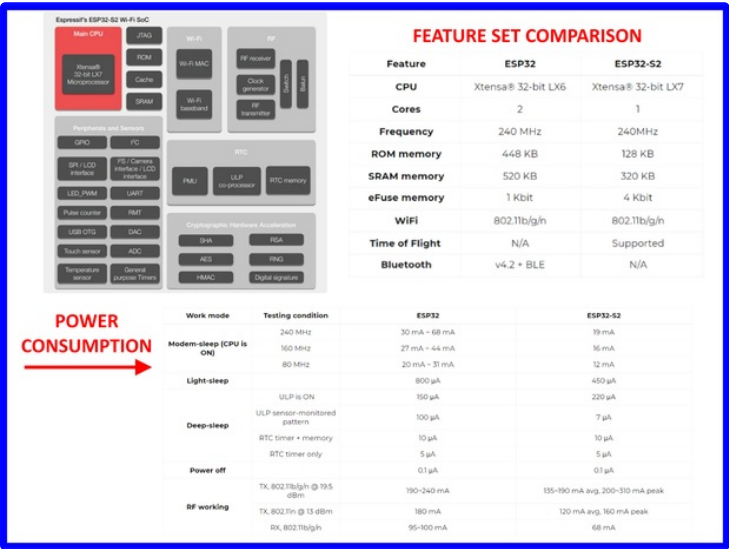
Four years ago, HackerBoxes brought you the new ESP32 System on Chip (SoC). The guide from that [HackerBox 0015 - Connect Everything](#) stands as a reminder to [Hack all the Things!](#) That guide included the following notice, copied here verbatim:

Notice: *The ESP32 is "the new hotness" and many of the tools and example programs are still in flux. It was no easy feat to get our hacky paws on these DevKits in such a large quantity because they are still mostly being provided to developers. Accordingly, this month the HackerBox call to "bring your hacker spirit" truly becomes a call to action. We have the opportunity to literally create some of the first examples, projects, and explanations for other hobbyists who will be flocking to the ESP32. We will also be doing some pioneering, finding some bugs, and we will need to be extra patient and understanding. Keep an eye on this instructable because it will evolve as our community collectively discovers, and hopefully creates, new opportunities with the Espressif ESP32.*

The spirit of this notice applies today for both of the new SoCs we're exploring: the **Espressif ESP32-S2** and the **Bouffalo Labs BL602 RISC-V**. So we repeat, "*the tools and example programs are still in flux*". Of course, four years after the notice above, our beloved ESP32 is widely used in a variety of systems and has many well-supported tools. We expect that will also become the case for the new variants of the ESP32 and for the numerous instances of RISC-V silicon that we will see hitting the market.

Venture Forth!

The tables in the image here illustrate the differences between the ESP32 and the new ESP32-S2. It is interesting to review the similar table in the [HackerBox 0015 Guide](#) linked above, which compares the ESP32 to the, then more common, 8266 and Arduino UNO.



Step 3: LILYGO ESP32-S2-WOOR Development Board

The LILYGO ESP32-S2-WOOR Development Board features the ESP32-S2-WROOM module ([datasheet](#)) along with 320KB of SRAM memory and 4MB of FLASH storage.

One of the more exciting features of the ESP32-S2 is integrated USB support. This development board still includes a CH340C USB chip which can be switched into (and out of) the path between the USB-C connector and the ESP32-S2. When the jumper switches are in USB mode, the CH340C chip is in play. When the jumper switches are in OTG mode, the USB-C connector is routed directly to the ESP32-S2.

Let's go...

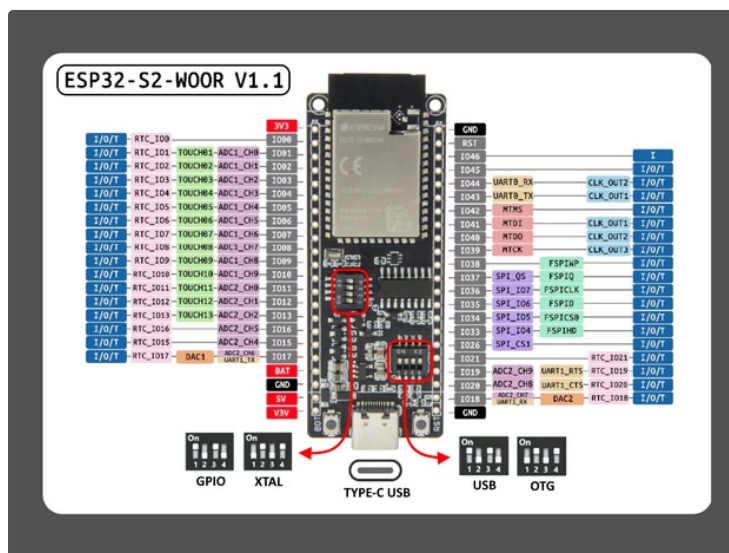
measurements.

Finally, the CPU will go into deep sleep for 60 seconds before waking up to do it again.

Low Power Operations

Plug the USB meter inline between the PC and the Development Board. Notice the power consumption. We observed 0.03A during the bootup execution boosting up to 0.07A with the Wi-Fi radio on and 0.000A in deep sleep.

Programming the ESP32-S2



For initial hook up, leave the header pins off and leave the jumper switches set to USB mode.

The connection to your PC can use a USB-C cable or a MicroUSB cable with the included Micro-to-C adapter.

Once connected, identify the COM port for the CH340 chip and open a serial monitor (the one built into the Arduino IDE works fine) on the identified COM port and set to 115.2Kbaud.

Hitting the RST button on the ESP32-S2 module will output startup information to the serial monitor.

The output will be followed by a scan of all 2.4GHz Wi-Fi access points in the area along with RSSI power

Arduino IDE: It is a bit of a hack, but the ESP32 files for the Arduino IDE can be swapped out for the ESP32-S2 files. This solution will surely be made more elegant in the future. For now, check out the instructions in [this video](#) from Andreas Spiess. (The actual tool setup starts at time offset 5:50.)

Official Espressif Toolchain: ESP-IDF for ESP32-S2

TinyUSB and CircuitPython: Deep Dive with Scott

Step 4: WSL - Run Linux on Windows

The Windows Subsystem for Linux (WSL) lets developers run a GNU/Linux environment directly on Windows, unmodified, without the overhead of a traditional virtual machine or dualboot setup. This includes most command-line tools, utilities, and applications.

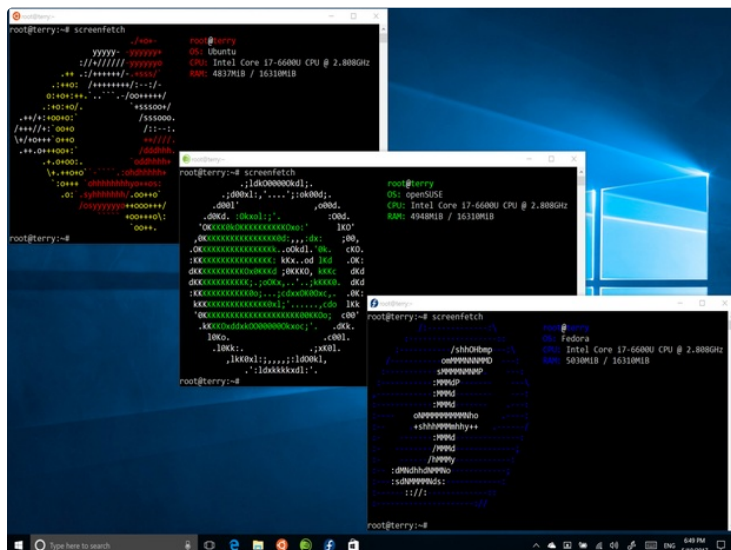
work with some security tools (of course you do) go with Kali.

Exploring Linux is an adventure worth taking for its own sake, but you will need it to run the RISC-V tools later.

Follow the instructions in the link. You can simply use WSL 1, skipping steps 2-5 which are for upgrading to WSL 2. Select the Linux distro of your choice. If you don't really care which, ubuntu is a nice choice. If you want to

Linux Tutorial for Beginners

Linux for Ethical Hackers (Kali Linux Tutorial)



Step 5: RISC-V

Engadget Video: [RISC-V is trying to launch an open-hardware revolution](#)

RISC-V ([Wikipedia](#)) is an open standard ISA (instruction set architecture) based on established reduced instruction set computer (RISC) principles. Unlike most other ISA designs, the RISC-V ISA is provided under open source licenses that do not require fees to use. A number of companies are offering or have announced RISC-V hardware, open source operating systems with RISC-V support are available and the instruction set is supported in several popular software toolchains.

The RISC-V project began in 2010 at the University of California, Berkeley along with many volunteer contributors not affiliated with the university. Unlike other academic designs which are typically optimized only for simplicity of exposition, the designers intended that the RISC-V instruction set be useable for practical computers.

CPU design requires design expertise in several

specialties: electronic digital logic, compilers, and operating systems. To cover the costs of such a team, commercial vendors of computer designs, such as ARM Holdings and MIPS Technologies charge royalties for the use of their designs, patents and copyrights. They also often require non-disclosure agreements before releasing documents that describe their designs' detailed advantages. In many cases, they never describe the reasons for their design choices.

RISC-V was started with a goal to make a practical ISA that was open-sourced, usable academically and in any hardware or software design without royalties. Also, the rationales for every part of the project are explained, at least broadly. The RISC-V authors are academics that have substantial experience in computer design. The RISC-V ISA is a direct development from a series of academic computer-design projects. It was originated in part to aid such projects.

[RISC-V International](#)



Step 6: Bouffalo Labs BL602 RISC-V Module

The DT-BL10 WiFi Development Board features the [Bouffalo Labs BL602 RISC-V IoT SoC](#). The BL602 SoC is a WiFi + BLE chipset, which may be used for low power consumption and high performance application development. The wireless subsystem includes 2.4GHz radio, Wi-Fi 802.11b/g/n, and BLE 5.0. The microcontroller subsystem includes a 32-bit RISC CPU with low power consumption, cache and memory. The power management unit controls the low power consumption mode. In addition, it also supports various security features. The external interfaces include SDIO, SPI, UART, I2C, IR remote, PWM, ADC, DAC, PIR and GPIO.

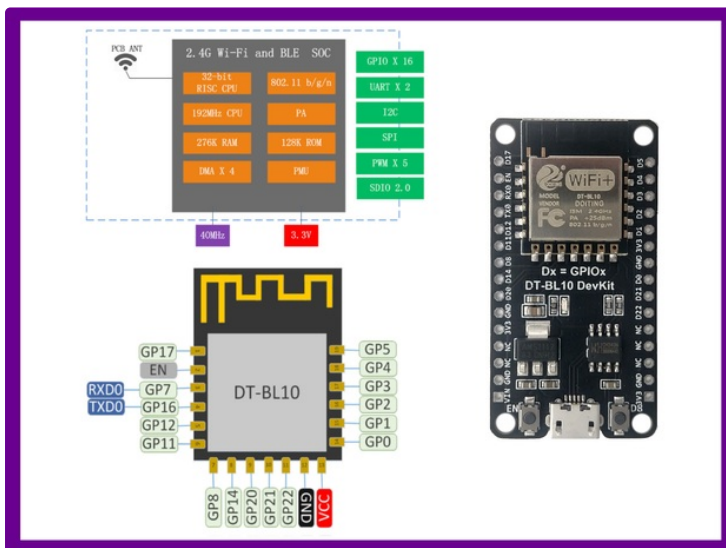
The BL602 is new RISC-V chip on the scene. The tools work well but are a little clumsy. In time, tools will be ported to more operating systems and quite possibly even integrated within common workspaces such as the Arduino IDE.

The first tools available run under Linux or in Windows leveraging the Windows Subsystem for Linux (WSL).

- Open a terminal in Linux
- Install make: `sudo apt-get install make`
- Install git: `sudo apt-get install git`
- Clone repo: `git clone https://github.com/SmartArduino/Doiting_BL.git`
- Change permissions: `chmod -R 777 ./Doiting_BL/bl_iot_sdk/toolchain/`
- More permissions: `find ./Doiting_BL/bl_iot_sdk/customer_app -name "genromap"|xargs chmod 777`
- Go to "Hello World" dir: `cd Doiting_BL/bl_IOT_SDK/customer_app/SDK_app_helloworld/`
- Execute script to compile: `./genromap`
- Go to flash_tool dir: `cd Doiting_BL\bl_iot_sdk\tools\flash_tool`
- Run flash_tool: (For example `./BLDevCube.exe` or `./BLDevCube`)
- If tool lacks permission to run, `chmod 777` on it
- Within tool: select BL602/604, UART, [correct COM port], 2Mbaud, 8N1, IoTKitA, 40M, Chip Erase: TRUE
- Click the checkboxes next to Partition Table, Boot2 Bin, and Firmware Bin
- For Partition Table, browse and select `partition_CFG_2m.toml`
- For Boot2 Bin, browse and select `blsp_boot2.bin`
- For Firmware Bin, browse to the `build_out` dir under `sdk_app_helloworld` and select `SDK_app_helloworld.bin`
- On the DT-BL10 board, press and hold D8, press EN, release EN, finally release D8
- Hit Create&Download button in flash tool
- A green success bar indicates flash was programmed
- Hit Open UART button in flash tool
- On the DT-BL10 board, hit the EN button
- Output "helloworld" should appear from serial output in the Log Window of flash tool

Congratulations on compiling and running your first RISC-V program

You can check the power consumption using the meter. We observed 0.000A while holding down the EN button and 0.035A when EN was released.



Step 7: Boost 3.3V From AA Batteries

These simple, yet efficient, switching power modules are based on the 2108 DC/DC boost converter ([datasheet](#)). They operate at 150 kHz and have an efficiency of around 85%.

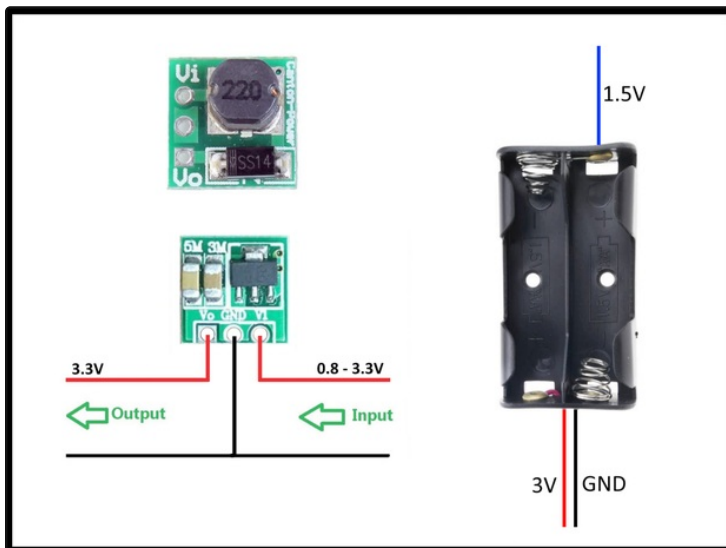
Input voltage: 0.8-3.3 V

Output voltage: 3.3V

Maximum output current depends on the input voltage:

- Vin of 0.8 V provides 10 mA max
- Vin of 1-1.5 V provides 50-110 mA
- Vin of 1.5-2 V provides 110-160 mA
- Vin of 2-3 V provides 160-400 mA
- Vin over 3 V provides 400-500 mA

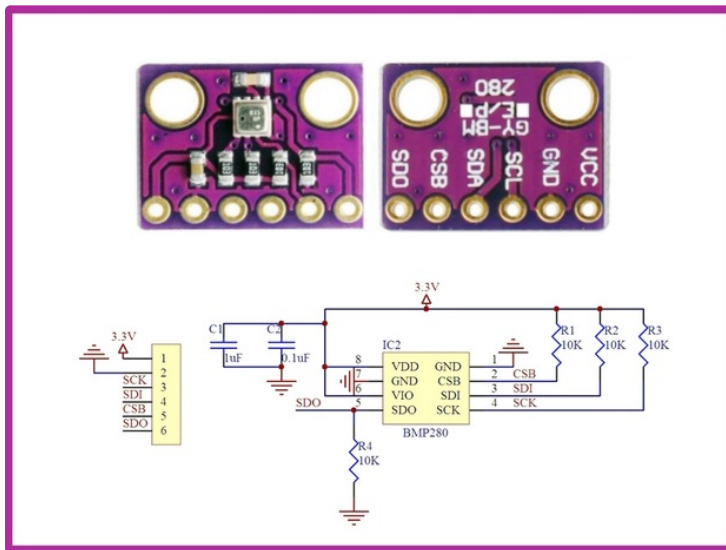
Theory of Operation



Step 8: BMP280 Temperature and Pressure Sensor

The BMP280 sensor module can measure air pressure and air temperature. The BMP280 digital pressure sensor is made by [Bosch](#). Bosch's BMP280 is an absolute barometric pressure sensor especially designed for mobile applications and measures both temperature and pressure.

Example project: [BMP280 with ESP32](#)



Step 9: Hack the Planet

We hope you are enjoying this month's HackerBox adventure into electronics and computer technology. Reach out and share your success in the comments below or other social media. Also, remember that you can email support@hackerboxes.com anytime if you have a question or need some help.

What's Next? Join the revolution. Live the HackLife. Get a cool box of hackable gear delivered right to your mailbox each month. Surf over to HackerBoxes.com and sign up for your monthly HackerBox subscription.



lol, this box gave me a flashback of the old boxes. Nondescript white box with a card showcasing the microcontroller inside.