HOW TO SETUP JABIT SOLO MINER (BITAXE V2.2)

STEPS:

1. Install and Setup standard ESP-IDF Toolchain – It's Free
2. Install Microsoft Visual Studio Code (VS Code) and ESP-IDF Plugin - It's Free
3. Establish Serial Connection with ESP32-S3 on JaBIT Solo Miner and setup up your configurations
4. Flash the Production Firmware to JaBIT Solo Miner (Bitaxe V2.2)
5. Start Mining.
6. For any issues of questions please join our discord bitaxe community. https://discord.gg/nRbcVjXm

1. *Installing and setting up standard ESP-IDF V4.4.4*

NOTE:
ESP-IDF requires some prerequisite tools to be installed so you can build firmware for supported chips. The prerequisite tools include Python, Git, cross-compilers, CMake and Ninja build tools.

Components of the installation

The installer deploys the following components:

- Embedded Python
- Cross-compilers
- OpenOCD
- CMake and Ninja build tools
- ESP-IDF

Follow the detailed instructions at :

For Windows:

https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/get-started/windows-setup.html

The easiest way to install ESP-IDF's prerequisites for windows is to download one of ESP-IDF Tools Installers at:

https://dl.espressif.com/dl/esp-idf/?idf=4.4

For Linux :

https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/get-started/linux-macos-setup.html

2. ***Installing Microsoft Visual Studio Code:***

   For Windows , Linux and Mac OS :

   For starters Visual Studio code provides an easier way of using ESP-IDF Toolchain.

   Follow instructions to install VSCode at :

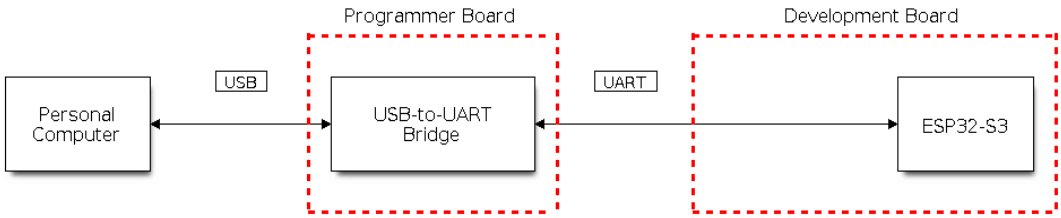
   https://github.com/espressif/vscode-esp-idf-extension/blob/master/docs/tutorial/install.md

3. ***Establish Serial Connection with ESP32-S3 on JaBIT Solo Miner***

   Currently JaBIT solo miner supports two ways of connecting to JaBIT solo Miner and flashing the firmware.

   1. Using ESP-PROG (USB-to_UART Bridge) and TAG-Connect TC2030-IDC-NL cable
   2. Using ESP-PROG (USB-to_UART Bridge) and Mini JTAG (J10) on JaBIT Solo Minier

External USB-to-UART Bridge



For detailed instructions:
https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/get-started/establish-serial-connection.html

4. ***Flash the Production Firmware (Nonce_checking Branch of ESP-Miner) to JaBIT Solo Miner (Bitaxe V2.2) .***

Note: Applies to both Windows and Linux OS

(a). After installing ESP-IDF V4.4 , VS Code and connecting your ESP-PROG to JaBIT Solo Miner you're now ready to Flash the production firmware to the Miner.

(b). Download the latest production firmware from the bitaxe community github repository and extract to folder.

- https://github.com/skot/ESP-Miner/tree/nonce_checking

OR

- https://github.com/developeralgo8888/ESP-Miner/tree/nonce_checking

(c). In VSCode select File → Open Folder → Select the Folder you extracted the nonce_checking files into.

(d). You use VS Code Terminal or command line to configure your project and flash the firmware .

1. Click on View --> Command Palette --> ESP-IDF: Device Configuration --> Device Target --> Select ESP-Miner-Nonce_Checking Folder -->esp32s3 --> ESP32-S3 chip(Via ESP-PROG)
2. Click on View --> Command Palette --> ESP-IDF: Device Configuration --> Device Port --> Enter your USB Port i.e COMxxx (Windows) or /dev/ttyUSBxxx (Linux)
3. Click on View --> Command Palette --> ESP-IDF: Device Configuration --> Flash baud rate --> 460800
4. Click on View --> Command Palette --> ESP-IDF: Device Configuration --> OpenOCD Config Files --> interface/ftdi/esp32_devkitj_v1.cfg,target/esp32s3.cfg
5. Click on View --> Command Palette --> ESP-IDF: Add VS Code configuration folder
6. Click on View --> Command Palette --> ESP-IDF: Open ESP-IDF Terminal --> it will open a terminal window in VS Code window and use the ESP-IDF Terminal to configure and flash the JaBIT Solo Miner. Type the following commands in the ESP-IDF terminal

Set the target:

```
idf.py set-target esp32s3
```

Use menuconfig to set the stratum server address/port and WiFi SSID/Password:

```
idf.py menuconfig
```

Set following parameters under **Stratum Configuration** **Options**, these will define the stratum server you connect to:

- Set `Stratum Address` to the stratum pool domain name. example "solo.ckpool.org"
- Set `Stratum Port` to the stratum pool port. example "3333"
- Set `Stratum username` to the stratum pool username
- Set `Stratum password` to the stratum pool password

Set following parameters under **Example Connection Configuration** Options:

- Set `WiFi SSID` to your target wifi network SSID.
- Set `Wifi Password` to the password for your target SSID

Build the project and flash it to the board, then run monitor tool to view serial output:

For Windows: (where xxx is your USB port number)

```
idf.py -p COMxxx flash monitor
```

For Linux:

```
idf.py -p /dev/ttyUSBxxx flash monitor
```

(To exit the serial monitor, type `Ctrl-]`.)

If the flashing of the production firmware is successful, you should see something similar to the output below and your small OLED screen should start displaying mining information.

```
developer@TestBox:~/Downloads/JABIT_BITAXE_FIRMWARE/ESP-Miner-nonce_checking$ idf.py -p /dev/ttyUSB2 flash monitor

Executing action: flash

Running ninja in directory /home/developer/Downloads/JABIT_BITAXE_FIRMWARE/ESP-Miner-nonce_checking/build

Executing "ninja flash"...

[1/5] cd /home/developer/Downloads/JABIT_BITAXE_FIRMWARE/ESP-Miner-nonce_checking/bui...eveloper/Downloads/JABIT_BITAXE_FIRMWARE/ESP-Miner-nonce_checking/build/esp-miner.bin

esp-miner.bin binary size 0xb0fb0 bytes. Smallest app partition is 0x100000 bytes. 0x4f050 bytes (31%) free.

[2/5] Performing build step for 'bootloader'

[1/1] cd /home/developer/Downloads/JABIT_BITAXE_FIRMWARE/ESP-Miner-nonce_checking/build/bootloader/esp-idf/esptool_py &&
```

```
/home/developer/.espressif/python_env/idf4.4_py3.10_env/bin/python /home/developer/esp/esp-idf/components/partition_table/check_sizes.py --offset 0x8000 bootloader 0x0 /home/developer/Downloads/JABIT_BITAXE_FIRMWARE/ESP-Miner-nonce_checking/build/bootloader/bootloader.bin

Bootloader binary size 0x5240 bytes. 0x2dc0 bytes (36%) free.

[2/3] cd /home/developer/esp/esp-idf/components/esptool_py && /usr/bin/cmake -D IDF_P...ing/build" -P /home/developer/esp/esp-idf/components/esptool_py/run_serial_tool.cmake

esptool.py esp32s3 -p /dev/ttyUSB2 -b 460800 --before=default_reset --after=hard_reset write_flash --flash_mode dio --flash_freq 80m --flash_size 2MB 0x0 bootloader/bootloader.bin 0x10000 esp-miner.bin 0x8000 partition_table/partition-table.bin

esptool.py v3.3.2

Serial port /dev/ttyUSB2

Connecting....

Chip is ESP32-S3

Features: WiFi, BLE

Crystal is 40MHz

MAC: 34:85:18:a6:f0:44

Uploading stub...

Running stub...

Stub running...

Changing baud rate to 460800

Changed.

Configuring flash size...

Flash will be erased from 0x00000000 to 0x00005fff...

Flash will be erased from 0x00010000 to 0x000c0fff...

Flash will be erased from 0x00008000 to 0x00008fff...

Compressed 21056 bytes to 13153...

Writing at 0x00000000... (100 %)

Wrote 21056 bytes (13153 compressed) at 0x00000000 in 0.5 seconds (effective 319.0 kbit/s)...

Hash of data verified.

Compressed 724912 bytes to 454173...

Writing at 0x00010000... (3 %)

Writing at 0x0001bfbb... (7 %)

Writing at 0x0002717c... (10 %)

Writing at 0x0002fecf... (14 %)

Writing at 0x00035f22... (17 %)

Writing at 0x0003b740... (21 %)
```

```
Writing at 0x00041f46... (25 %)
Writing at 0x00047fca... (28 %)
Writing at 0x0004da25... (32 %)
Writing at 0x000530d5... (35 %)
Writing at 0x00058944... (39 %)
Writing at 0x0005e2b6... (42 %)
Writing at 0x00063b4f... (46 %)
Writing at 0x00068b88... (50 %)
Writing at 0x0006dceb... (53 %)
Writing at 0x00072ba4... (57 %)
Writing at 0x00078014... (60 %)
Writing at 0x0007d4b5... (64 %)
Writing at 0x000830e7... (67 %)
Writing at 0x0008890a... (71 %)
Writing at 0x0008e570... (75 %)
Writing at 0x000947de... (78 %)
Writing at 0x0009a08f... (82 %)
Writing at 0x000a2156... (85 %)
Writing at 0x000a94ab... (89 %)
Writing at 0x000b12f2... (92 %)
Writing at 0x000b6d72... (96 %)
Writing at 0x000bcde8... (100 %)
Wrote 724912 bytes (454173 compressed) at 0x00010000 in 10.4 seconds (effective 557.6 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 103...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.0 seconds (effective 512.8 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
Executing action: monitor
Running idf_monitor in directory /home/developer/Downloads/JABIT_BITAXE_FIRMWARE/ESP-Miner-nonce_checking
```

```
Executing "/home/developer/.espressif/python_env/idf4.4_py3.10_env/bin/python
/home/developer/esp/esp-idf/tools/idf_monitor.py -p /dev/ttyUSB2 -b 115200 --toolchain-prefix
xtensa-esp32s3-elf- --target esp32s3 /home/developer/Downloads/JABIT_BITAXE_FIRMWARE/ESP-Miner-
nonce_checking/build/esp-miner.elf -m
'/home/developer/.espressif/python_env/idf4.4_py3.10_env/bin/python' '/home/developer/esp/esp-
idf/tools/idf.py' '-p' '/dev/ttyUSB2'"...

--- idf_monitor on /dev/ttyUSB2 115200 ---

--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

x<x( boo

Build:Mar 27 2021

rst:0x1 (POWERON),boot:0x2a (SPI_FAST_FLASH_BOOT)

SPIWP:0xee

mode:DIO, clock div:1

load:0x3fce3808,len:0x1668

load:0x403c9700,len:0xbcc

load:0x403cc700,len:0x2fac

entry 0x403c9954

I (25) boot: ESP-IDF v4.4.4-dirty 2nd stage bootloader

I (25) boot: compile time 05:08:53

I (25) boot: chip revision: v0.1

I (27) boot_comm: chip revision: 1, min. bootloader chip revision: 0

I (34) boot.esp32s3: Boot SPI Speed : 80MHz

I (39) boot.esp32s3: SPI Mode       : DIO

I (44) boot.esp32s3: SPI Flash Size : 2MB

I (48) boot: Enabling RNG early entropy source...

I (54) boot: Partition Table:

I (57) boot: ## Label            Usage          Type ST Offset   Length

I (65) boot:  0 nvs              WiFi data        01 02 00009000 00006000

I (72) boot:  1 phy_init         RF data          01 01 0000f000 00001000

I (80) boot:  2 factory          factory app      00 00 00010000 00100000

I (87) boot: End of partition table

I (91) boot_comm: chip revision: 1, min. application chip revision: 0

I (98) esp_image: segment 0: paddr=00010020 vaddr=3c090020 size=1746ch ( 95340) map

I (124) esp_image: segment 1: paddr=00027494 vaddr=3fc97970 size=05340h ( 21312) load

I (129) esp_image: segment 2: paddr=0002c7dc vaddr=40374000 size=0383ch ( 14396) load

I (133) esp_image: segment 3: paddr=00030020 vaddr=42000020 size=80e3ch (527932) map

I (233) esp_image: segment 4: paddr=000b0e64 vaddr=4037783c size=10128h ( 65832) load
```

```
I (256) boot: Loaded app from partition at offset 0x10000

I (256) boot: Disabling RNG early entropy source...

I (267) cpu_start: Pro cpu up.

I (267) cpu_start: Starting app cpu, entry point is 0x40375480

0x40375480: call_start_cpu1 at /home/developer/esp/esp-
idf/components/esp_system/port/cpu_start.c:148


I (0) cpu_start: App cpu up.

I (281) cpu_start: Pro cpu start user code

I (281) cpu_start: cpu freq: 160000000

I (281) cpu_start: Application information:

I (284) cpu_start: Project name:     esp-miner

I (289) cpu_start: App version:      1

I (294) cpu_start: Compile time:     Jun 22 2023 05:08:47

I (300) cpu_start: ELF file SHA256:  36f5140952dc22c5...

I (306) cpu_start: ESP-IDF:          v4.4.4-dirty

I (311) heap_init: Initializing. RAM available for dynamic allocation:

I (318) heap_init: At 3FCA0DF8 len 00048918 (290 KiB): D/IRAM

I (325) heap_init: At 3FCE9710 len 00005724 (21 KiB): STACK/DRAM

I (331) heap_init: At 3FCF0000 len 00008000 (32 KiB): DRAM

I (338) heap_init: At 600FE000 len 00002000 (8 KiB): RTCRAM

I (344) spi_flash: detected chip: gd

I (348) spi_flash: flash io: dio

W (352) spi_flash: Detected size(16384k) larger than the size in the binary image header(2048k).
Using the size in the binary image header.

I (366) sleep: Configure to isolate all GPIO pins in sleep state

I (372) sleep: Enable automatic switching of GPIO sleep configuration

I (379) cpu_start: Starting scheduler on PRO CPU.

I (0) cpu_start: Starting scheduler on APP CPU.

I (400) miner: Welcome to the bitaxe!
```

IMPORTANT :

NOTE 1:

JaBIT Solo Miner can mine in 2 ways :

- If you use Testing Mode then you must use **I2C_test Firmware** with **cgminer-bitaxe** otherwise it will not work. You will get ***0 chips*** found. If you try to use ***cgminer-bitaxe with the current installed production firmware that ships with JaBIT Solo Miner it will not work***.

---

1. Production Mode:

JaBIT comes installed with production firmware (From Nonce_Checking Branch) but it will not connect or mine because the WIFI and Bitcoin Address are from our testing network before we ship it to you.

What you need:

(a). Firmware: **Nonce_Checking** branch ---- is the production firmware

- Download , **Configure** , Compile and flash the Nonce_Checking branch firmware to JaBIT Solo Miner and you are ready to start mining .
- You don't need anything else. Follow the instructions above on how to setup JaBIT Solo Miner.
- Download from **our production firmware** repos :
    - https://github.com/skot/ESP-Miner/tree/nonce_checking OR
    - https://github.com/developeralgo8888/ESP-Miner/tree/nonce_checking

---

2. Testing Mode:

Uses **Cgminer-bitaxe 4.12.1** (This is a **Modified Cgminer** to work with **JaBIT Solo Miner (Bitaxe V2.2))** installed on an External PC with our testing firmware **I2C_test** branch.

In this Testing Mode you need.

(a). Firmware: **I2C_test** --- is only meant for testing/debugging with cgminer-bitaxe

(b). Cgminer-bitaxe

(c). **FTDI** UART Serial USB

- From the our **I2C_test** Branch of the github Repo:

    - https://github.com/skot/ESP-Miner/tree/i2c_test OR
    - https://github.com/developeralgo8888/ESP-Miner/tree/i2c_test

    - Download, compile and flash the JaBIT Solo Miner with the I2C_test Firmware.

    - Now you can download and compile the **modified cgminer-bitaxe** version for Mining with JaBIT solo Miner (Bitaxe V2.2)

    - https://github.com/skot/cgminer-bitaxe OR
    - https://github.com/developeralgo8888/cgminer-bitaxe

    - This **I2C_test** Firmware and **cgminer-bitaxe** is meant for **testing and debugging** functionalities of JaBIT Solo Miner and proving that it works as expected.

How to Connect your FTDI UART Serial USB to JaBIT Solo Miner :

UART USB side		**JaBIT Solo Miner(BitAxe v2.2) side**
VCC	------>	1V8
GND	------>	GND
TXD	------>	RXI
RXD	------>	TXO
RTS	------>	**RST** (on the JaBIT Solo Miner(Bitaxe V2.2) PCB)
CTS	------>	Not Connected

Note: Standard cgminer-kanoi version uses **CBUS1** pin on UART Serial USB to reset the BM1397 Chip before it starts mining or receiving mining commands.

Our Modified cgminer-bitaxe uses **RTS** Pin on UART Serial USB to reset the BM1397 before it starts mining or receiving mining commands. RTS pin is standard on most of FTDI UART Serial USB in the market.

NOTE 2: This cables might change in the future as we improve the JaBIT Solo Miner

Here are some links to the cables you will need if you don't already have them. We use the same cables and ESP_PROG board to program and flash ESP32-S3 Controller that JaBIT Solo Miner uses.

1. **ESP-PROG (ESP32 Programmer -- ESP-Prog JTAG Development Board)** -- This is the UART Serial Bridge to your JaBIT Solo Miner . Note : Prices vary from US $19 to US$35 depending on where you buy it.

   https://www.amazon.com/ESP-Prog-Development-Debugging-Downloader-Compatible/dp/B09LC49DFR/ref=sr_1_6?crid=2YXPWT13MB1T6&keywords=ESP32-PROG&qid=1687459439&sprefix=esp32-prog%2Caps%2C94&sr=8-6

2. **TC2030-IDC-NL** cable (Connects ESP-PROG to JaBIT TC2030 pins) Note: Price is around 32 Euros or US$34

   https://www.tag-connect.com/product/tc2030-idc-nl

   SKU: TC2030-IDC-NL --- “No Legs” 6-pin Plug-of-Nails™ Cable fitted with a 6-pin 0.1” pitch ribbon connector.

3. **If you are in EU you might need to replace the power cable with two prongs cable**. The included power cable is for North America.

   The Power adapter (PSU) itself (Input 100 --- 240V @50/60Hz 1.6A) will work for both EU and North America

   Here is simple power cable for EU countries

   https://www.amazon.com/Kentek-Prongs-IEC320-International-European/dp/B07KS5LP5J/ref=sr_1_18?crid=3B1XPN0NDAD6O&keywords=EU+Power+cable&qid=1687460471&sprefix=eu+power+cable%2Caps%2C110&sr=8-18

If you search Amazon or Ebay or any other site online in EU , you should be able to get them pretty quickly.