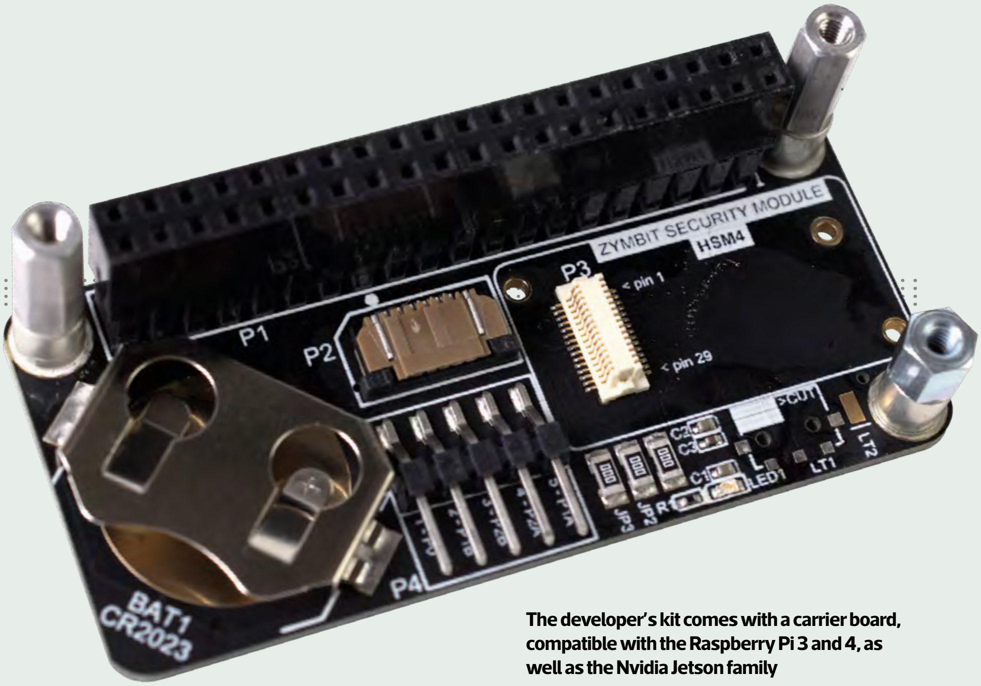
REVIEW

Zymbit HSM6



The developer's kit comes with a carrier board, compatible with the Raspberry Pi 3 and 4, as well as the Nvidia Jetson family

Your average computer hobbyist usually wouldn't care much about hardware security modules, unless the presence of one prevented them from doing something. Zymbit, though, has been trying to convince hobbyists that hardware security modules are of interest for some time – and its latest attempt piggybacks onto the growing popularity of cryptocurrencies.

The Zymbit HSM6 is a direct successor to the earlier HSM4, which in turn is a designed-for-manufacture module based on the same technology as its Zymkey4.

All three devices have one feature in common: an Arm Cortex-M0 microcontroller coupled with a Microchip ATECC608 secure element, which is accessible through application programming interfaces (APIs) for C, C++ and Python.

The Zymkey4 is a stick-like board that attaches to the general-purpose input/output (GPIO) header of a Raspberry Pi 3 or 4, or an Nvidia Jetson Xavier NX or Jetson Nano single-board computer. The HSM4 and HSM6, by contrast, are tiny modules whose electronics are entirely encased in resin to protect them from physical attack. The only exposed part is a high-density connector on the underside.

The actual HSM6 is tiny, and entirely encased in resin bar a single connector

If you're designing a product, a tiny module such as this one is easy to integrate into it. If you're a hobbyist, or someone experimenting at the very start of a product design cycle, it's not so easy.

Thankfully, Zymbit sells a kit that bundles a single module with a compact carrier board. You pop the module and a battery – CR2032, not CR2023 as labelled – onto the board, then the board onto the GPIO header of your Raspberry Pi or Nvidia Jetson.

Getting the software is quick and easy, although it falls into the same trap as many other products for Linux-based devices, encouraging the user to pipe a script, sight-unseen, through the shell as the root user. Thankfully, the script doesn't do anything nasty. It checks your system, installs various prerequisites and adds Zymbit's repository to the apt package manager for future updates.

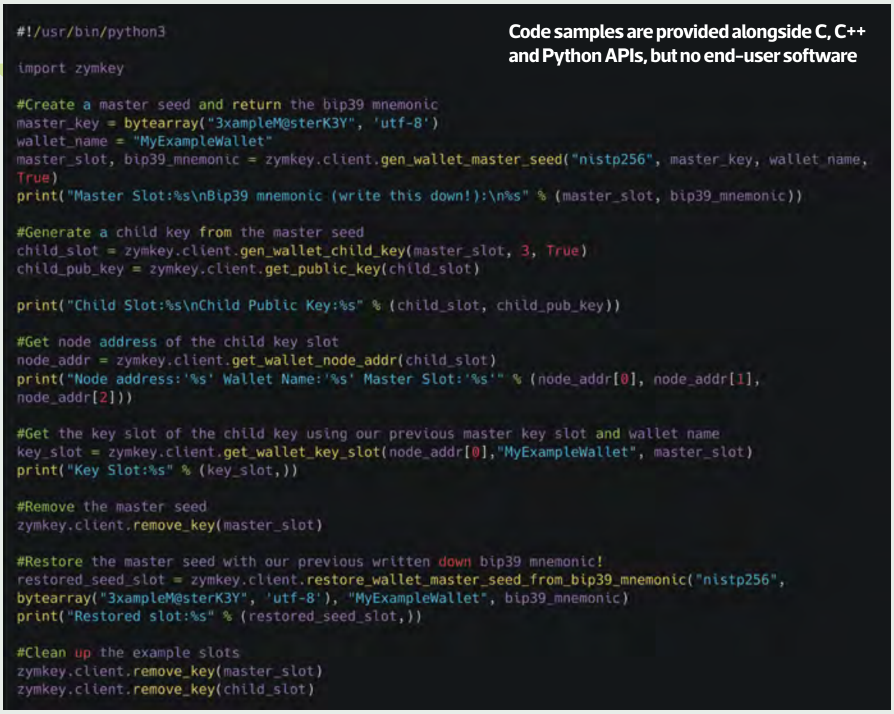
Once installed, the HSM6, like the HSM4 before it, fires up in 'developer mode'. There's a reason why this mode is the default – 'production mode' irrevocably binds the module to the current host.

If you switch from developer to production mode, there's no way to go backwards. If you want to switch to a different host device, you'll need to pick up a new HSM6 module.

That's not a criticism, but a key aspect of the HSM6. It aims to provide as close to complete security as possible. Keys are generated, stored and used directly on the HSM6 without being exposed to the host system. Follow the guide for encrypting data, or even your entire storage device, using the HSM6 and you've got data that will work on that single Raspberry Pi or Nvidia Jetson board and no other.

Encryption, decryption, signing and verification are only some of the tricks up the HSM6's compact sleeve. Another key feature is a true random number generator (TRNG), which can provide high-quality entropy to your applications – or, with a little ingenuity, the host operating system as well.

The high-density connector is surprisingly robust for its small size



```
#!/usr/bin/python3

import zymkey

#Create a master seed and return the bip39 mnemonic
master_key = bytearray("3xampleM@sterK3Y", 'utf-8')
wallet_name = "MyExampleWallet"
master_slot, bip39_mnemonic = zymkey.client.gen_wallet_master_seed("nistp256", master_key, wallet_name,
True)
print("Master Slot:%s\nBip39 mnemonic (write this down!):\n%s" % (master_slot, bip39_mnemonic))

#Generate a child key from the master seed
child_slot = zymkey.client.gen_wallet_child_key(master_slot, 3, True)
child_pub_key = zymkey.client.get_public_key(child_slot)

print("Child Slot:%s\nChild Public Key:%s" % (child_slot, child_pub_key))

#Get node address of the child key slot
node_addr = zymkey.client.get_wallet_node_addr(child_slot)
print("Node address:'%s' Wallet Name:'%s' Master Slot:'%s'" % (node_addr[0], node_addr[1],
node_addr[2]))

#Get the key slot of the child key using our previous master key slot and wallet name
key_slot = zymkey.client.get_wallet_key_slot(node_addr[0],"MyExampleWallet", master_slot)
print("Key Slot:%s" % (key_slot,))

#Remove the master seed
zymkey.client.remove_key(master_slot)

#Restore the master seed with our previous written down bip39 mnemonic!
restored_seed_slot = zymkey.client.restore_wallet_master_seed_from_bip39_mnemonic("nistp256",
bytearray("3xampleM@sterK3Y", 'utf-8'), "MyExampleWallet", bip39_mnemonic)
print("Restored slot:%s" % (restored_seed_slot,))

#Clean up the example slots
zymkey.client.remove_key(master_slot)
zymkey.client.remove_key(child_slot)
```

Code samples are provided alongside C, C++ and Python APIs, but no end-user software

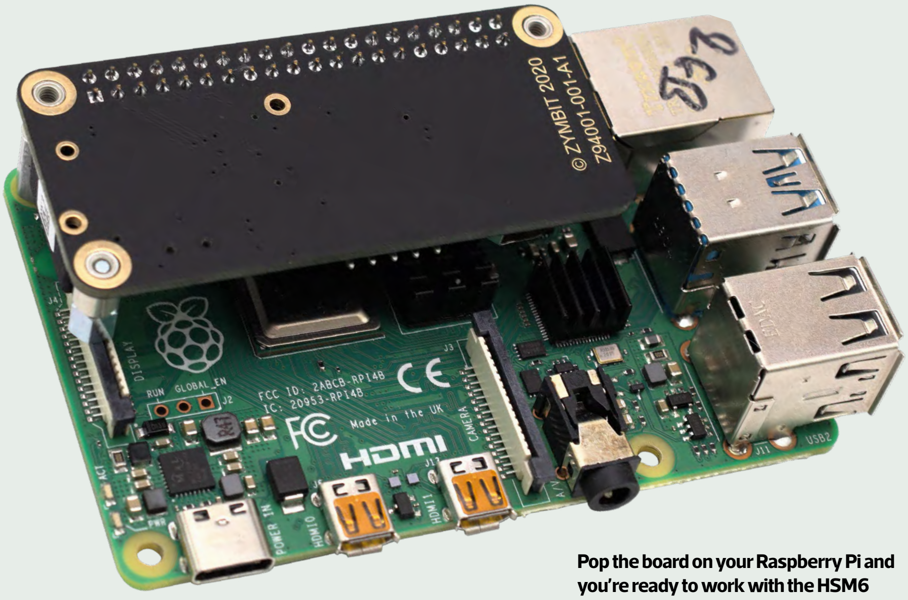
The HSM6 also includes three elements designed to protect against physical attack, on top of the binding to a single host. A power quality monitor flags attempts to mess with the power supply, an accelerometer can be triggered on shocks or changes in orientation, and a two-ring perimeter monitoring circuit can be wired to trigger on a device's case being opened or damaged.

It's easy to play with the latter as well. It's brought out to a simple pin header on the carrier board, and wires bundled with the kit let you simulate the monitoring circuits being broken. Be careful though – one of the possible responses to physical attack is a self-destruct mode, which leaves the module permanently out of action.

So far, the features of the HSM4 and HSM6 have been identical. What the HSM6 adds is support for handling the keys for a cryptocurrency wallet, allowing for BIP32/39/44 wallet generation while keeping the keys hidden from the host operating system.

Python source code is provided, showing how to take advantage of the HSM6's new digital wallet API, but that's as far as Zymbit goes. There's no user-facing software supplied, and no integration with common wallet software. Unless the HSM6 enjoys considerable success in the market, that's unlikely to change. This means the cryptocurrency support could be of interest to developers, but it's not much use to end users simply looking to keep their crypto coins out of the hands of ne'er-do-wells.

The biggest issue with using the HSM6 as a hobbyist, though, is the price. The developer kit is priced at $155 US, with additional modules costing $125 each (around £112 and £90 respectively, ex VAT). For security wonks, the feature list makes the asking price worth it; for the merely curious, though, it's too much. For cryptocurrency enthusiasts, a dedicated hardware wallet is similarly priced and far more usable. The HSM6 is available from **zymbit.com** now.



Pop the board on your Raspberry Pi and you're ready to work with the HSM6

NEWS IN BRIEF

Custom chips for under $10K

Efabless is now offering custom chip production to everyone for $9,750 US (around £7,000 ex VAT) per project. The firm earlier partnered with Google and SkyWater to offer open-source hardware projects the opportunity to build their designs as actual silicon chips.

For the cash, customers – who can be anyone from well-heeled hobbyists to commercial ventures – receive either 100 QFN or 300 WCSP-packaged chips of their design on a 130nm process node. Additional chips are available at an extra cost, the company has confirmed. More information on the scheme, dubbed chipIgnite, is available at **efabless.com**

DEEP DIVE HOW NVIDIA DLSS AND AMD FSR WORK

CUSTOM PC

THE BEST-SELLING MAG FOR PC HARDWARE, OVERCLOCKING, GAMING & MODDING / ISSUE 217

EXCLUSIVE

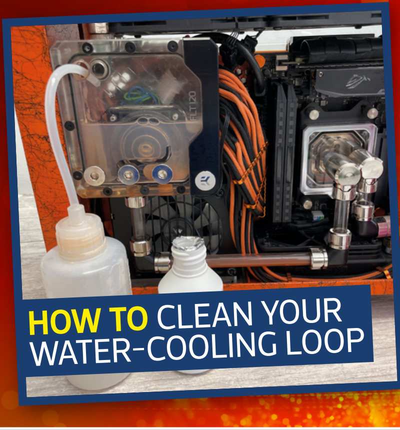
BEAT THE SCALPERS

BUILD A **£1,099** PC WITH PARTS YOU CAN ACTUALLY BUY

- EVGA GEFORCE RTX 3060
- AMD RYZEN 5 5600X
- 16GB OF RAM
- 1TB PCI-E SSD
- LIQUID COOLING
- AND MORE

GUARANTEED FOR FIRST 100 READERS

GROUP TESTS
CPU AIR COOLERS
WIRELESS GAMING MICE



HOW TO CLEAN YOUR WATER-COOLING LOOP

OCTOBER 2021 / £5.99

9 771740 744059 10

Raspberry Pi PRESS

40 YEARS OF THE PC WE TALK TO IBM