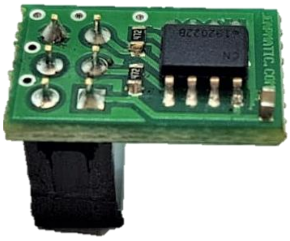
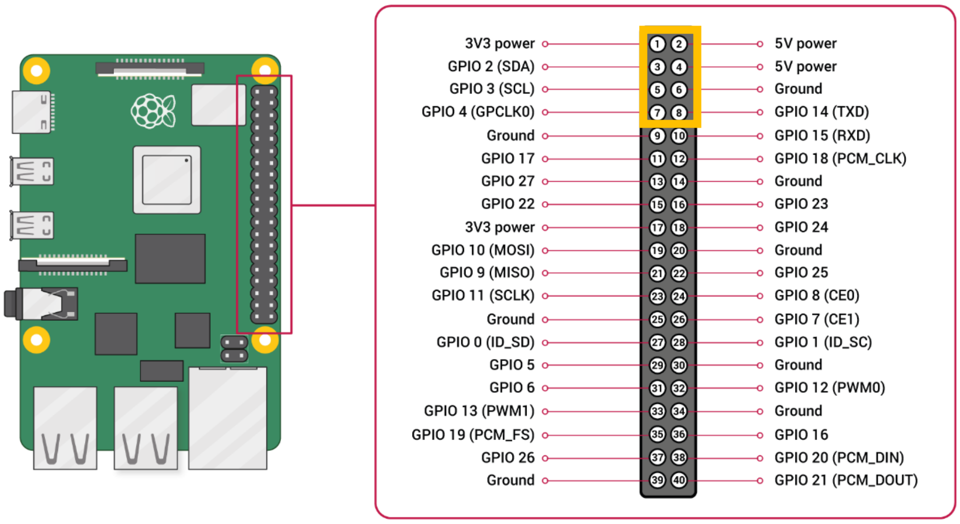
HATL01C

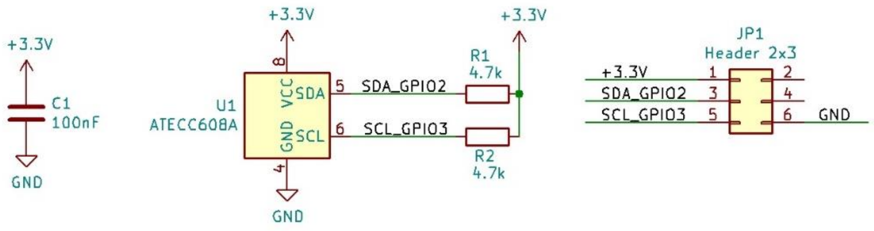
Crypto -module for Raspberry Pi.

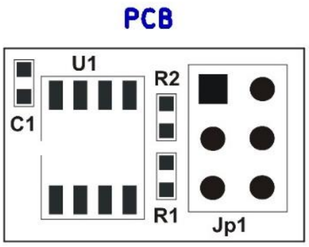


Keep your software safe from stealing with this MaticControl crypto module. Most microcontrollers are not designed to protect against snoopers, but a crypto-authentication chip can be used to lock away private keys securely. Once the private key is saved inside the chip, it can't be read out, all you can do is send it challenge-response queries. That means that even if someone gets hold of your hardware and can read back the firmware, they won't be able to extract it! Place it on pins 1-8. And this is all you have to do on the hardware.


3V3 power 1 2 5V power
GPIO 2 (SDA) 3 4 5V power
GPIO 3 (SCL) 5 6 Ground
GPIO 4 (GPCLK0) 7 8 GPIO 14 (TXD)
Ground 9 10 GPIO 15 (RXD)
GPIO 17 11 12 GPIO 18 (PCM_CLK)
GPIO 27 13 14 Ground
GPIO 22 15 16 GPIO 23
3V3 power 17 18 GPIO 24
GPIO 10 (MOSI) 19 20 Ground
GPIO 9 (MISO) 21 22 GPIO 25
GPIO 11 (SCLK) 23 24 GPIO 8 (CE0)
Ground 25 26 GPIO 7 (CE1)
GPIO 0 (ID_SD) 27 28 GPIO 1 (ID_SC)
GPIO 5 29 30 Ground
GPIO 6 31 32 GPIO 12 (PWM0)
GPIO 13 (PWM1) 33 34 Ground
GPIO 19 (PCM_FS) 35 36 GPIO 16
GPIO 26 37 38 GPIO 20 (PCM_DIN)
Ground 39 40 GPIO 21 (PCM_DOUT)

Electrical Scheme of the module:

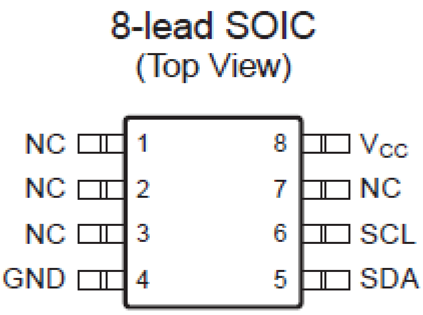




Note: This MaticControl module covers the pins for: 5V / 3,3V power supply, I2C, GPIO14 (RXD) and GPIO14 (TXD). If you want to use them, we offer modules (HATs) that provide access to these pins via separate connectors on the top of the board.

See more at LeapMatic.com

The ATECC608 is the latest crypto-auth chip from Microchip and to make working with the it as easy as possible, we've put it on a PCB . This allows you to use it with Raspberry Pi or other similarly equipped boards **without needing to solder.**



ATECC608 uses I2C to send/receive commands. It will work with 3.3V or 5V power/logic micros, so it's ready to get to work with a range of development boards. Once you 'lock' the chip with your details, you can use it for ECDH and AES-128 encrypt/decrypt/signing. There's also hardware support for random number generation, and SHA-256/HMAC hash functions to greatly speed up a slower micro's cryptography commands.

For our surprise this chip does not have a public datasheet, but it is compatible with the ATECC508 earlier version which does, so please refer to that complete datasheet as well as the ATECC608 summary sheet.

Some quick notes from the official Microchip datasheet:

Applications

The ATECC508A device is a member of the Microchip CryptoAuthentication™ family of crypto engine authentication devices with highly secure hardware-based key storage.
The ATECC508A device has a flexible command set that allows use in many applications, including the following:

- **Network/IoT Node Protection** - Authenticates node IDs, ensures the integrity of messages, and supports key agreement to create session keys for message encryption.
- **Anti-Counterfeiting** - Validates that a removable, replaceable, or consumable client is authentic. Examples of clients could be system accessories, electronic daughter cards, or other spare parts. It can also be used to validate a software/firmware module or memory storage element.
- **Protecting Firmware or Media** - Validates code stored in flash memory at boot to prevent unauthorized modifications, encrypt downloaded program files as a common broadcast, or uniquely encrypt code images to be usable on a single system only.
- **Storing Secure Data** - Stores secret keys for use by crypto accelerators in standard microprocessors. Programmable protection is available using encrypted/authenticated reads and writes.
- **Checking User Password** - Validates user-entered passwords without letting the expected value become known, maps memorable passwords to a random number, and securely exchanges password values with remote systems.

Device Features

The ATECC508A includes an EEPROM array which can be used for storage of up to 16 keys, certificates, miscellaneous read/write, read-only or secret data, consumption logging, and security configurations. Access to the various sections of memory can be restricted in a variety of ways and then the configuration can be locked to prevent changes.

The ATECC508A features a wide array of defense mechanisms specifically designed to prevent physical attacks on the device itself, or logical attacks on the data transmitted between the device and the system. Hardware restrictions on the ways in which keys are used or generated provide further defense against certain styles of attack.

Access to the device is made through a standard I2C Interface at speeds of up to 1 Mb/s. The interface is compatible with standard Serial EEPROM I2C interface specifications. The device also supports a Single- Wire Interface (SWI), which can reduce the number of GPIOs required on the system processor, and/or reduce the number of pins on connectors. If the Single-Wire Interface is enabled, the remaining pin is available for use as a GPIO, an authenticated output or tamper input.

Using either the I2C or Single-Wire Interface, multiple ATECC508A devices can share the same bus, which saves processor GPIO usage in systems with multiple clients such as different color ink tanks or multiple spare parts, for example.

Each ATECC508A ships with a guaranteed unique 72-bit serial number. Using the cryptographic protocols supported by the device, a host system or remote server can verify a signature of the serial number to prove that the serial number is authentic and not a copy. Serial numbers are often stored in a standard Serial EEPROM; however, these can be easily copied with no way for the host to know if the serial number is authentic or if it is a clone.

The ATECC508A can generate high-quality FIPS random numbers and employ them for any purpose, including usage as part of the device's crypto protocols. Because each random number is guaranteed to be essentially unique from all numbers ever generated on this or any other device, their inclusion in the protocol calculation ensures that replay attacks (i.e. re-transmitting a previously successful transaction) will always fail.

Cryptographic Operation

The ATECC508A implements a complete asymmetric (public/private) key cryptographic signature solution based upon Elliptic Curve Cryptography and the ECDSA signature protocol. The device features hardware acceleration for the NIST standard P256 prime curve and supports the complete key life cycle from high quality private key generation, to ECDSA signature generation, ECDH key agreement, and ECDSA public key signature verification.

The hardware accelerator can implement such asymmetric cryptographic operations from ten to onethousand times faster than software running on standard microprocessors, without the usual high risk of key exposure that is endemic to standard microprocessors.

The device is designed to securely store multiple private keys along with their associated public keys and certificates. The signature verification command can use any stored or an external ECC public key. Public keys stored within the device can be configured to require validation via a certificate chain to speed up subsequent device authentications.

Random private key generation is supported internally within the device to ensure that the private key can never be known outside of the device. The public key corresponding to a stored private key is always returned when the key is generated and it may optionally be computed at a later time.

The ATECC508A also supports a standard hash-based challenge-response protocol in order to simplify programming. In its most basic instantiation, the system sends a challenge to the device, which combines that challenge with a secret key via the MAC, HMAC or SHA commands and then sends the response back to the system. The device uses a SHA-256 cryptographic hash algorithm to make that combination so that an observer on the bus cannot derive the value of the secret key, but preserving the ability of a recipient to verify that the response is correct by performing the same calculation with a stored copy of the secret on the recipient's system.

Due to the flexible command set of the ATECC508A, these basic operation sets (i.e. ECDSA signatures, ECDH key agreement and SHA-256 challenge-response) can be expanded in many ways. Using the GenDig command, the values in other slots can be included in the response digest or signature, which provides an effective way of proving that a data read really did come from the device, as opposed to being inserted by a man-in-the-middle attacker. This same command can be used to combine two keys with the challenge, which is useful when there are multiple layers of authentication to be performed. In a host-client configuration where the host (for instance, a mobile phone) needs to verify a client (for instance, an OEM battery), there is a need to store the secret in the host in order to validate the response from the client. The CheckMac command allows the device to securely store the secret in the host system and hides the correct response value from the pins, returning only a yes or no answer to the system.

Finally, the hash combination of a challenge and secret key can be kept on the device and XORed with the contents of a slot to implement an encrypted Read command, or it can be XORed with encrypted input data to implement an encrypted Write command.

All hashing functions are implemented using the industry-standard SHA-256 secure hash algorithm, which is part of the latest set of high-security cryptographic algorithms recommended by various government agencies and cryptographic experts. The ATECC508A employs full-sized 256-bit secret keys to prevent any kind of exhaustive attack.