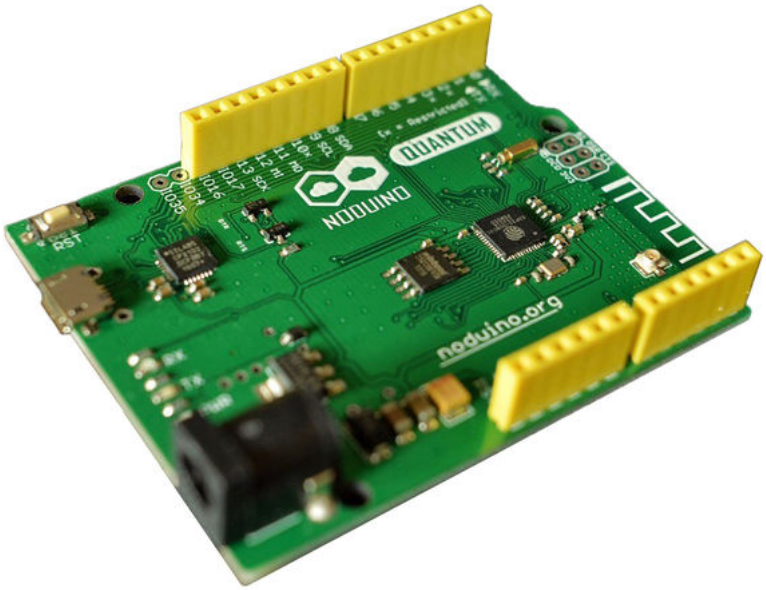
# Noduino Quantum

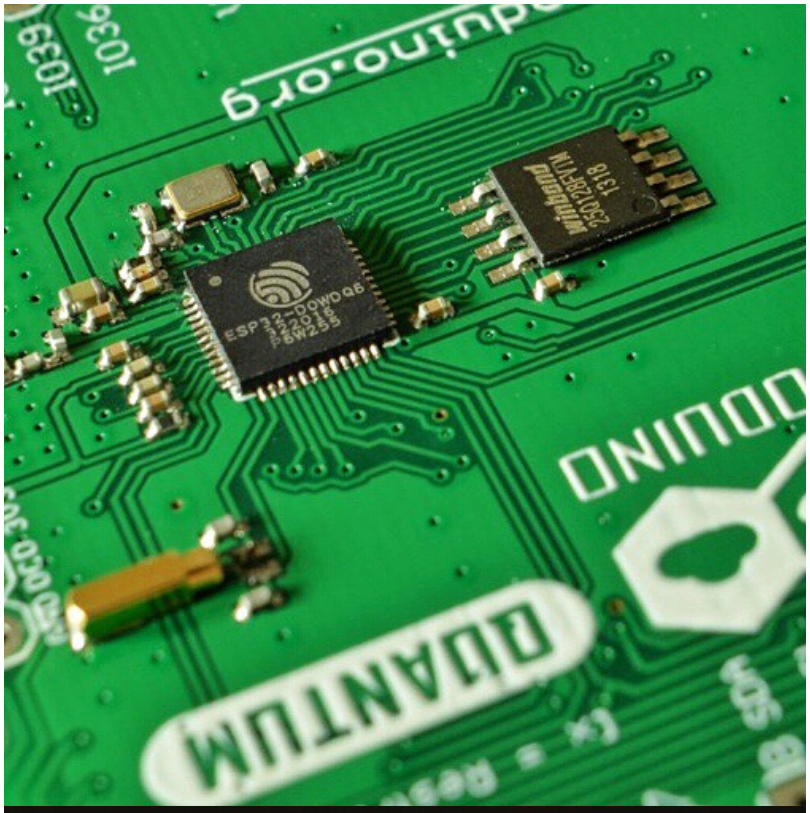
来自Jack's Lab

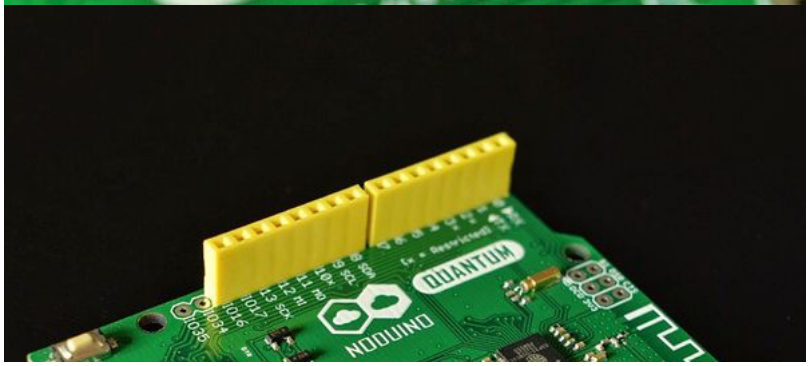
## 目录

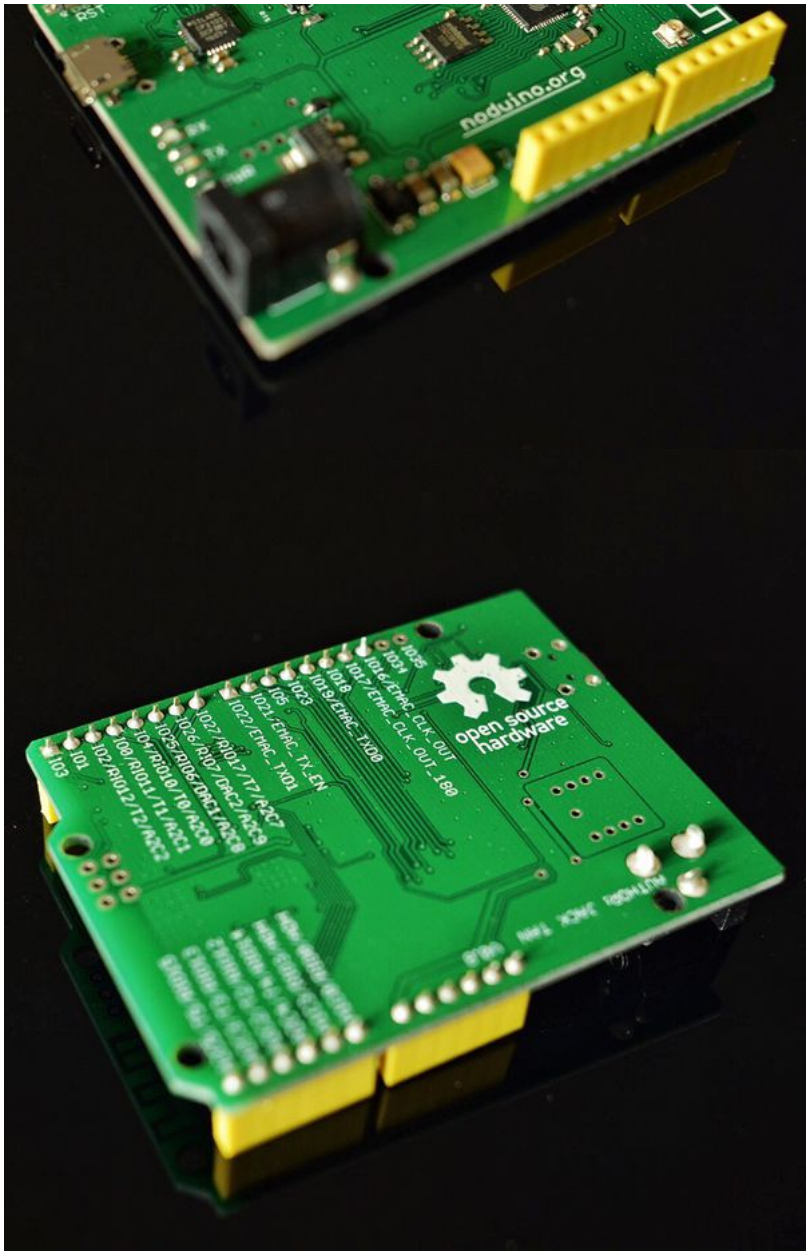


### 1 Overview


QUANTUM
NODUINO
noduino.org


noduino.org
QUANTUM
NODUINO


QUANTUM
NODUINO




- CP2102 USB to UART Chip
- ESP32 Bluetooth and WiFi SoC
- 40MHz Crystal (±10ppm, ±10ppm)
- 16MB SPI Flash
- 5V - 12V Power Supply
- FreeRTOS

- UART Baud rate is 115200

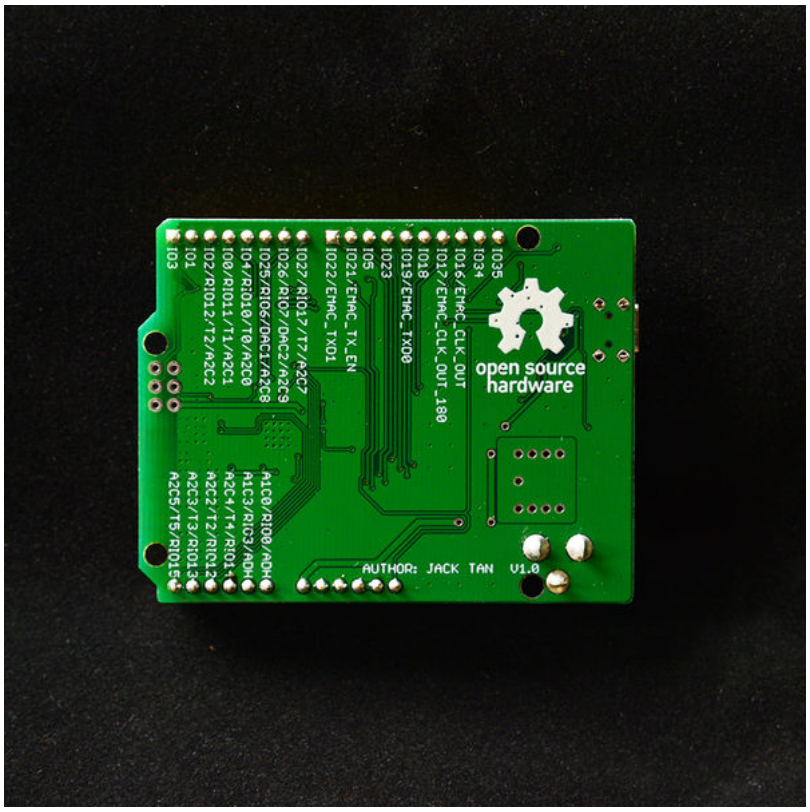
## 2 Flash mode

We use two types of 16MB SPI flash:

- Winbond W25Q128xxxx, Support mode: QIO / QOUT / DIO / DOUT; 80MHz / 40MHz
- MXIC 25L128xxx, Support mode: DIO / DOUT; 80MHz / 40MHz

# 3 Pin Map

We place the PIN map on the back of the board:




You guys can refer to this doc for more details: http://www.espressif.com/sites/default/files/documentation/esp32_chip_pin_list_en_0.pdf

IO_MUX

| Pin No. | Pin Name | Pin Name | Pin Name | Power Domain | Analog Function1 | Analog Function2 | Analog Function3 | RTC_GPIO | Function1 | Type | Function2 | Type | Function3 | Type | Function4 | Type | Function5 | Type | Function6 | Type | Drive Strength (2'd2: 24mA) | At Reset | After Reset |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | VDDA | | | VANA in | | | | | | | | | | | | | | | | | | | |
| 2 | | LNA_IN | | VANA in | | | | | | | | | | | | | | | | | | | |
| 3 | VDD3P3 | | | VANA in | | | | | | | | | | | | | | | | | | | |
| 4 | VDD3P3 | | | VANA in | | | | | | | | | | | | | | | | | | | |
| 5 | | SENSOR_VP | | VRTC | ADC_H | ADC1_CH0 | | RTC_GPIO0 | GPIO36 | I | | | GPIO36 | I | | | | | | | | | ie=0 |
| 6 | | SENSOR_CAPP | | VRTC | ADC_H | ADC1_CH1 | | RTC_GPIO1 | GPIO37 | I | | | GPIO37 | I | | | | | | | | | ie=0 |
| 7 | | SENSOR_CAPN | | VRTC | ADC_H | ADC1_CH2 | | RTC_GPIO2 | GPIO38 | I | | | GPIO38 | I | | | | | | | | | ie=0 |
| 8 | | SENSOR_VN | | VRTC | ADC_H | ADC1_CH3 | | RTC_GPIO3 | GPIO39 | I | | | GPIO39 | I | | | | | | | | | ie=0 |
| 9 | | CHIP_PU | | VRTC | | | | | | | | | | | | | | | | | | | |
| 10 | | VDET_1 | | VRTC | | ADC1_CH6 | | RTC_GPIO4 | GPIO34 | I | | | GPIO34 | I | | | | | | | | | ie=0 |
| 11 | | VDET_2 | | VRTC | | ADC1_CH7 | | RTC_GPIO5 | GPIO35 | I | | | GPIO35 | I | | | | | | | | | ie=0 |
| 12 | | 32K_XP | | VRTC | XTAL_32K_P | ADC1_CH4 | TOUCH9 | RTC_GPIO9 | GPIO32 | I/O/T | | | GPIO32 | I/O/T | | | | | | | 2'd2 | | ie=0 |
| 13 | | 32K_XN | | VRTC | XTAL_32K_N | ADC1_CH5 | TOUCH8 | RTC_GPIO8 | GPIO33 | I/O/T | | | GPIO33 | I/O/T | | | | | | | 2'd2 | | ie=0 |
| 14 | | | GPIO25 | VRTC | DAC_1 | ADC2_CH8 | | RTC_GPIO6 | GPIO25 | I/O/T | | | GPIO25 | I/O/T | | | | | EMAC_RXD0 | I | 2'd2 | | ie=0 |
| 15 | | | GPIO26 | VRTC | DAC_2 | ADC2_CH9 | | RTC_GPIO7 | GPIO26 | I/O/T | | | GPIO26 | I/O/T | | | | | EMAC_RXD1 | I | 2'd2 | | ie=0 |
| 16 | | | GPIO27 | VRTC | | ADC2_CH7 | TOUCH7 | RTC_GPIO17 | GPIO27 | I/O/T | | | GPIO27 | I/O/T | | | | | EMAC_RX_DV | I | 2'd2 | | ie=1 |
| 17 | | | MTMS | VRTC | | ADC2_CH6 | TOUCH6 | RTC_GPIO16 | MTMS | I0 | HSPICLK | I/O/T | GPIO14 | I/O/T | HS2_CLK | O | SD_CLK | I0 | EMAC_TXD2 | O | 2'd2 | | ie=1 |
| 18 | | | MTDI | VRTC | | ADC2_CH5 | TOUCH5 | RTC_GPIO15 | MTDI | I1 | HSPIQ | I/O/T | GPIO12 | I/O/T | HS2_DATA2 | I1/O/T | SD_DATA2 | I1/O/T | EMAC_TXD3 | O | 2'd2 | wpd, ie=1 | wpd, ie=1 |
| 19 | VDD3P3_RTC | | | VRTC supply in | | | | | | | | | | | | | | | | | | | |
| 20 | | | MTCK | VRTC | | ADC2_CH4 | TOUCH4 | RTC_GPIO14 | MTCK | I1 | HSPID | I/O/T | GPIO13 | I/O/T | HS2_DATA3 | I1/O/T | SD_DATA3 | I1/O/T | EMAC_RX_ER | I | 2'd2 | | ie=1 |
| 21 | | | MTDO | VRTC | | ADC2_CH3 | TOUCH3 | RTC_GPIO13 | MTDO | O/T | HSPICS0 | I/O/T | GPIO15 | I/O/T | HS2_CMD | I1/O/T | SD_CMD | I1/O/T | EMAC_RXD3 | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 22 | | | GPIO2 | VRTC | | ADC2_CH2 | TOUCH2 | RTC_GPIO12 | GPIO2 | I/O/T | HSPIWP | I/O/T | GPIO2 | I/O/T | HS2_DATA0 | I1/O/T | SD_DATA0 | I1/O/T | | | 2'd2 | wpd, ie=1 | wpd, ie=1 |
| 23 | | | GPIO0 | VRTC | | ADC2_CH1 | TOUCH1 | RTC_GPIO11 | GPIO0 | I/O/T | CLK_OUT1 | O | GPIO0 | I/O/T | | | | | EMAC_TX_CLK | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 24 | | | GPIO4 | VRTC | | ADC2_CH0 | TOUCH0 | RTC_GPIO10 | GPIO4 | I/O/T | HSPIHD | I/O/T | GPIO4 | I/O/T | HS2_DATA1 | I1/O/T | SD_DATA1 | I1/O/T | EMAC_TX_ER | O | 2'd2 | wpd, ie=1 | wpd, ie=1 |
| | | | GPIO20 | VSDIO | | | | | GPIO20 | I/O/T | | | GPIO20 | I/O/T | | | | | | | 2'd2 | | ie=1 |
| 25 | | | GPIO16 | VSDIO | | | | | GPIO16 | I/O/T | | | GPIO16 | I/O/T | HS1_DATA4 | I1/O/T | U2RXD | I1 | EMAC_CLK_OUT | O | 2'd2 | | ie=1 |
| 26 | VDD_SDIO | | | VSDIO supply out/in | | | | | | | | | | | | | | | | | | | |
| 27 | | | GPIO17 | VSDIO | | | | | GPIO17 | I/O/T | | | GPIO17 | I/O/T | HS1_DATA5 | I1/O/T | U2TXD | O | EMAC_CLK_OUT_180 | O | 2'd2 | | ie=1 |
| 28 | | | SD_DATA_2 | VSDIO | | | | | SD_DATA2 | I1/O/T | SPIHD | I/O/T | GPIO9 | I/O/T | HS1_DATA2 | I1/O/T | U1RXD | I1 | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 29 | | | SD_DATA_3 | VSDIO | | | | | SD_DATA3 | I0/O/T | SPIWP | I/O/T | GPIO10 | I/O/T | HS1_DATA3 | I1/O/T | U1TXD | O | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 30 | | | SD_CMD | VSDIO | | | | | SD_CMD | I1/O/T | SPICS0 | I/O/T | GPIO11 | I/O/T | HS1_CMD | I1/O/T | U1RTS | O | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 31 | | | SD_CLK | VSDIO | | | | | SD_CLK | I0 | SPICLK | I/O/T | GPIO6 | I/O/T | HS1_CLK | O | U1CTS | I1 | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 32 | | | SD_DATA_0 | VSDIO | | | | | SD_DATA0 | I1/O/T | SPIQ | I/O/T | GPIO7 | I/O/T | HS1_DATA0 | I1/O/T | U2RTS | O | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 33 | | | SD_DATA_1 | VSDIO | | | | | SD_DATA1 | I1/O/T | SPID | I/O/T | GPIO8 | I/O/T | HS1_DATA1 | I1/O/T | U2CTS | I1 | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 34 | | | GPIO5 | VIO | | | | | GPIO5 | I/O/T | VSPICS0 | I/O/T | GPIO5 | I/O/T | HS1_DATA6 | I1/O/T | | | EMAC_RX_CLK | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 35 | | | GPIO18 | VIO | | | | | GPIO18 | I/O/T | VSPICLK | I/O/T | GPIO18 | I/O/T | HS1_DATA7 | I1/O/T | | | | | 2'd2 | | ie=1 |
| 36 | | | GPIO23 | VIO | | | | | GPIO23 | I/O/T | VSPID | I/O/T | GPIO23 | I/O/T | HS1_STROBE | I0 | | | | | 2'd2 | | ie=1 |
| 37 | VDD3P3_CPU | | | VIO supply in | | | | | | | | | | | | | | | | | | | |
| 38 | | | GPIO19 | VIO | | | | | GPIO19 | I/O/T | VSPIQ | I/O/T | GPIO19 | I/O/T | U0CTS | I1 | | | EMAC_TXD0 | O | 2'd2 | | ie=1 |
| 39 | | | GPIO22 | VIO | | | | | GPIO22 | I/O/T | VSPIWP | I/O/T | GPIO22 | I/O/T | U0RTS | O | | | EMAC_TXD1 | O | 2'd2 | | ie=1 |
| 40 | | | U0RXD | VIO | | | | | U0RXD | I1 | CLK_OUT2 | O | GPIO3 | I/O/T | | | | | | | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 41 | | | U0TXD | VIO | | | | | U0TXD | O | CLK_OUT3 | O | GPIO1 | I/O/T | | | | | EMAC_RXD2 | I | 2'd2 | wpu, ie=1 | wpu, ie=1 |
| 42 | | | GPIO21 | VIO | | | | | GPIO21 | I/O/T | VSPIHD | I/O/T | GPIO21 | I/O/T | | | | | EMAC_TX_EN | O | 2'd2 | | ie=1 |
| 43 | VDDA | | | VANA in | | | | | | | | | | | | | | | | | | | |
| 44 | | XTAL_N | | VANA | | | | | | | | | | | | | | | | | | | |
| 45 | | XTAL_P | | VANA | | | | | | | | | | | | | | | | | | | |
| 46 | VDDA | | | VANA | | | | | | | | | | | | | | | | | | | |
| 47 | | CAP2 | | VANA | | | | | | | | | | | | | | | | | | | |
| 48 | | CAP1 | | VANA | | | | | | | | | | | | | | | | | | | |
| Total Number | 8 | 14 | 26 | | | | | | | | | | | | | | | | | | | | |

# 4 Quick Start

## 4.1 USB2UART

Quantum use the CP2102 USB to UART chip, you need to install the driver firstly. Accessing following url to get your driver:

- http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx

The default baud rate is 115200

## 4.2 Arduino

We have pushed the support of Quantum board into arduino-esp32. Following steps to try the arduino quickly:

- Install Arduino IDE
- Go to Arduino IDE installation directory:

```
$ cd /PATH/TO/Arduino
$ cd hardware
$ mkdir espressif
$ cd espressif
$ git clone git://github.com/espressif/arduino-esp32.git esp32
$ cd esp32/tools
$ python get.py
```

- Restart Arduino IDE

Example: ESP32 RFID

## 4.3 ESP IDF

### 4.3.1 Linux

Please refer to: https://github.com/icamgo/esp-idf/blob/master/docs/linux-setup.rst

Simple steps:

```
$ sudo apt-get install git wget make libncurses-dev flex bison gperf python python-serial
$ wget https://dl.espressif.com/dl/xtensa-esp32-elf-linux32-1.22.0-59.tar.gz
$ mkdir -p toolchain
$ tar zxf xtensa-esp32-elf-linux32-1.22.0-59.tar.gz -C toolchain
$ export PATH=$PATH:`pwd`/toolchain/xtensa-esp32-elf/bin
$
$ git clone --recursive git://github.com/icamgo/esp-idf.git
$ export IDF_PATH=`pwd`/esp-idf
$ cd esp-idf/examples/09_onchip_sensor
$ make menuconfig
$ make flash
```

### 4.3.2 MAC OS

Please refer to: https://github.com/icamgo/esp-idf/blob/master/docs/macos-setup.rst

### 4.3.3 Windows

Please refer to: https://github.com/icamgo/esp-idf/blob/master/docs/windows-setup.rst

## 4.4 Auto Reset

You need to press the RST buttom after uploading the firmware into flash. If you guys do not like to do this please patch the /path/to/esp-idf/components/esptool_py/esptool/esptool.py :

```
diff --git a/esptool.py b/esptool.py
index 755f4cb..ff92c91 100755
--- a/esptool.py
+++ b/esptool.py
@@ -197,6 +197,12 @@ class ESPLoader(object):
                 + '\xc0'
         self._port.write(buf)

+    def reset_to_app(self):
+        self._port.setDTR(False)
+        self._port.setRTS(True)
+        time.sleep(0.05)
+        self._port.setRTS(True)
+
     """ Calculate checksum of a blob, as it is defined by the ROM """
     @staticmethod
     def checksum(data, state=ESP_CHECKSUM_MAGIC):
@@ -1421,7 +1427,6 @@ def dump_mem(esp, args):
         sys.stdout.flush()
     print 'Done!'

-
 def write_flash(esp, args):
     """Write data to flash
     """
@@ -1503,6 +1508,7 @@ def write_flash(esp, args):
     if args.verify:
         print 'Verifying just-written flash...'
         verify_flash(esp, args, header_block)
+    esp.reset_to_app()


 def image_info(args):
```

Then Quantum can reset to run your app automatically after uploading the firmware into flash

## 5 Turtorial

- ESP32 Smartconfig Support WeChat Airkiss by default
- ESP32 JTAG Using a FT2232H breakout board to debug the kernel/core through JTAG/OpenOCD/GDB
- ESP32 Boot
- ESP32 RTC External Wakeup
- ESP32 RFID MFRC522 RFID (SPI interface), Using the Arduino IDE to compile and upload...
- ESP32 ADC ESP32 Onchip SAR ADC Turtorial
- ESP32 DAC ESP32 Onchip SAR DAC Turtorial
- ESP32 Partical Sharp GP2Y1010AU0F Particle Sensor Turtorial
- ESP32 Onchip Sensor read the onchip Temperature sensor and Hall sensor Turtorial
- ESP32 SSD1306 OLED Screen Turtorial (I2C interface)
- ESP32 Camera OV7725 Camera Turtorial

## 6 Peripherals

- ESP32 GPIO
- ESP32 I2C
- ESP32 SPI
- ESP32 CAN
- ESP32 I2S
- ESP32 RMII
- ESP32 Onchip Sensor read the onchip Temperature sensor and Hall sensor
- ESP32 ADC ESP32 Onchip SAR ADC
- ESP32 DAC ESP32 Onchip DAC
- ESP32 PWM LED PWM Controller
- ESP32 RMT IR Remote Controller

- ESP32 RTC
- ESP32 ULP

- ESP32 TSL2561 TSL2561 Digital Luminosity/Lux/Light Sensor (I2C interface)
- ESP32 BH1750 BH1750 Digital Light Sensor (I2C interface)
- ESP32 BMP180 BMP180 Barometric Pressure/Temperature/Altitude Sensor (I2C interface)
- ESP32 BMP085 BMP085 Barometric Pressure/Temperature/Altitude Sensor (I2C interface)

- ESP32 SHT2x SHT2X Digital Humidity & Temperature Sensor (I2C interface)
- ESP32 DHT21 DHT21(AM2301) Digital Temperature & Humidity Sensor

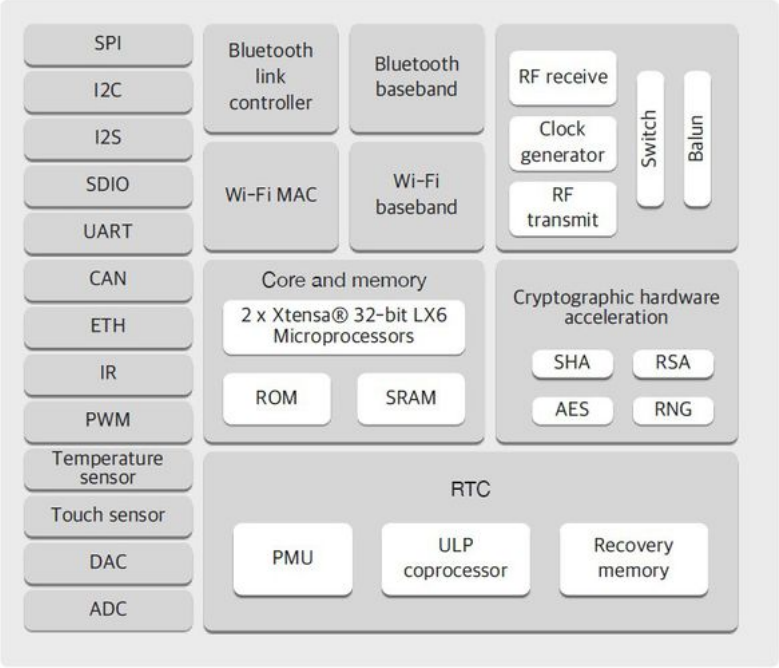
- ESP32 DHT11 DHT11 Digital Humidity & Temperature Sensor
- ESP32 PT1000 Using a 18-bit ADC MCP3421 (I2C interface)

- ESP32 Partical Sharp GP2Y1010AU0F Particle Sensor support

- ESP32 PCF8563 PCF8563 I2C RTC Chip (I2C interface)
- ESP32 SSD1306 OLED Screen (I2C interface)
- ESP32 RFID MFRC522 RFID (SPI interface)

## 7 ESP32 Arch

ESP32 block diagram:




ESP32 use two LX6 core which ISA is xtensa.

- Xtensa LX6 Core: http://ip.cadence.com/uploads/533/Cadence_Tensillica_Xtensa_LX6_ds-pdf
- Xtensa Instruction Set Architecture: http://0x04.net/~mwk/doc/xtensa.pdf

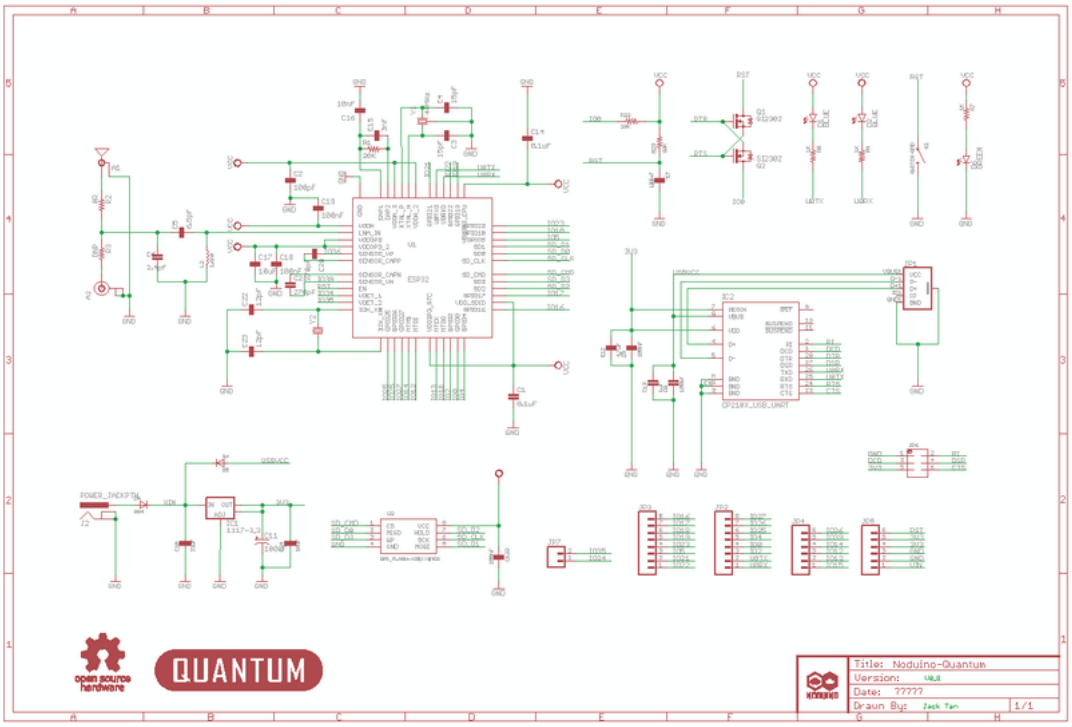
We plan to write the document of the xtensa architecture like the MIPS (http://wiki.jackslab.org/MIPS_%E4%BD%93%E7%B3%BB%E7%BB%93%E6%9E%84%E7%9B%B8%E5%85%B3%E6%96%87%E9%9B%86%E6%B1% or SPARC (http://www.jackslab.org/?skill-type=%E4%BD%93%E7%B3%BB%E7%BB%93%E6%9E%84)

The Xtensa instruction set is designed to meet the diverse requirements of dataplane processing. This 32-bit architecture features a compact 16- and 24-bit instruction set with modeless switching for maximum power efficiency and performance. The base instruction set has 80 RISC instructions and includes a 32-bit ALU, up to 64 general-purpose 32-bit registers, and six special-purpose registers. Using this instruction set, you can expect significant code size reductions that result in higher code density and better power dissipation.

- Xtensa GPR and ABI
- Xtensa Instruction Set
- Xtensa Exception
- Xtensa Memory

- ESP32 SPR
- ESP32 core isa

## 8 Hardware




## 9 Reference

For more information please refer to

- Noduino