TWILIO electric imp

Electric Imp imp006

Internet of Things Module

Datasheet

Preliminary

Contents

1. Introduction

The Electric Imp imp006 is a highly flexible Internet of Things (IoT) MCU with fully customizable wireless Internet connectivity. It incorporates impOS™, which delivers hardware access, device management and a secure virtual machine environment in which customers' over-the-air (OTA) delivered application firmware executes. The imp006 is made available in a number of variations: each is intended for a specific connectivity configuration.

2. Key Features

- Custom STM32F423ZHJ6I Microcontroller:
  - Integrated real-time operating system, impOS™
  - 96MHz, 32-bit ARM Cortex M4F core
  - 240KB application RAM
  - 512KB application Flash area
- Rich peripheral support:
  - 54x General Purpose I/O
  - 4x SPI channels
  - 7x UART channels
  - 3x I^2C channels
  - 8x PWM channels
  - ADC/DAC support
  - DFSDM digital audio input support
- Low power:
  - impOS™-managed idle mode with full RAM preservation and wake on timer or GPIO, typically 40μA
  - Ultra-low power 6μA sleep mode with RTC and GPIO wake
- Secure by default, secure for life:
  - OS, network and security stack updates provided by Twilio for the lifetime of the device
  - MCU ships with full hardware protection enabled
  - Independently certified to the UL2900-2-2 cybersecurity for industrial control systems standard
- Flexible wireless support:
  - Choose cellular only, WiFi only, or cellular and WiFi
  - Global cellular connectivity powered by Twilio Super SIM
  - Murata single- and dual-band WiFi + BLE modules supported, with impOS™-maintained BLE stack
  - Quectel BG96 cellular radio provides global Cat-M, 2G and NB connectivity
  - Full support for BG96 GNSS including cloud-assisted warm start
- Works in partnership with impCloud™ platform:
  - Secure, managed cloud connection via TLS1.2
  - Mutual authentication and state of the art EDH (Elliptic-curve Diffie-Hellman) forward secrecy
  - Secure, failsafe OTA OS and application upgrades
  - Cloud-side companion agent for each device
- System-managed QSPI flash for OS and application use:
  - 64Mbit and 128Mbit chips supported
  - 32Mbit to 96Mbit available for application use
- Multiple provisioning options:
  - End-user BlinkUp™ for easy, secure setup, either optically or via BLE where available
  - In-factory provisioning for zero-touch end-user install

3. imp006 Variants

The imp006 supports multiple network technologies by decoupling the network hardware from the MCU. Customers can connect a range of network subsystems to the imp006 and control which of these is used to deliver Internet connectivity. Customers can tailor the network technologies they support to their application’s particular use-cases.

The imp006 has three variants according to the wireless components it may be connected to:

Variant	Device ID Starts With	WiFi Module	Cellular Module
imp006a	`0x600a`	Murata 1MW Dual-band: 2.4/5GHz	Quectel BG96, optional
imp006b	`0x600b`	Murata 1DX Single-band: 2.4GHz	Quectel BG96, optional
imp006c	`0x6000`	None	Quectel BG96, mandatory

Note 1 Both Murata 1xx parts also provide BLE . This is accessible in code via the imp API.
Note 2 The BG96 also provides GNSS support. This is accessible in code via the imp API.
Note 3 Local network access (UDP) is only supported by WiFi-equipped variants: imp006a and imp006b.

Electric Imp provides customers with an appropriate encrypted firmware for the variant they have chosen. This firmware should be installed on an external SPI flash. On its first boot in the customer’s factory, the STM32F423ZHJ6I will verify the firmware and copy it to its internal flash.

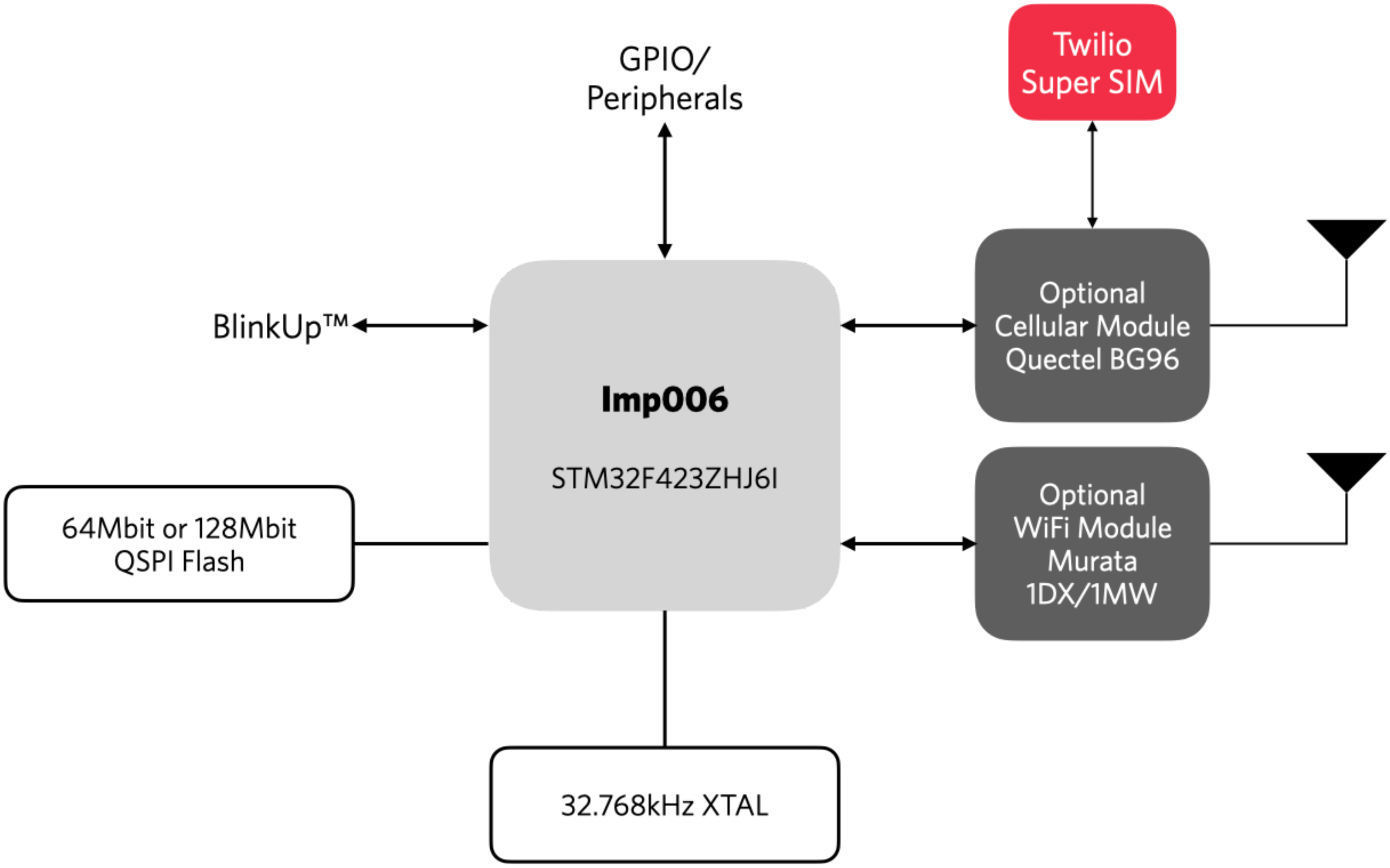
From this point, the module is an imp006 of the selected variant and cannot be changed to another variant. All subsequent boots execute the internal firmware.

4. imp006 Features and Peripheral Counts

MCU	
Microcontroller	Based on the ST STM32F423ZHJ6I
Size	10x10mm
CPU	32-bit ARM Cortex M4F @ 96MHz
Application RAM	240KB
Application flash	512KB
IO and Peripherals	
GPIO Pins	54
SPI	Four channels
UART	Seven channels
I^2C	Three channels
PWM	Eight channels
Analog/Digital	
ADC Pins	Ten
DAC Pins	Two
Fixed-frequency DAC	Yes
DFSDM* in	Yes
*Connectivity***	
WiFi	Optional Murata 1DX single- or 1MW dual-band WiFi module
Cellular	Optional Quectel BG96 CAT-M1/NB-IoT module
BLE	Optional, included with WiFi module
GNSS	Optional, included with WiFi module Supports GPS, Galilio, Glonass, Beidou
Antenna	Radio-specific
Approvals	Recommended radios have modular approvals
Implementation	
Application Storage	512KB (approx. 20,000+ lines of code excluding comments)
Minimum Design Requirement	64Mbit SPI flash, phototransistor, bi-color LED, 32kHz xtal for low-power mode support, four-layer PCB
Optional Design Features	Application rescue pin

* Digital Filter for Sigma-Delta Modulators
** At least one networking technology is required: cellular only, WiFi only, or cellular/WiFi.

5. System Diagram



6. Component Requirements

Your design will require the following in addition to the imp006 module.

6.1 BlinkUp™ Reception Circuitry

The required circuit is straightforward: you just need a phototransistor and a resistor.

Your phototransistor must be able to sense visible light. Do not use a phototransistor with a visible-light filter. The value of the bias resistor used in the circuit will depend greatly on your mechanical design and on the opto-electrical components you have selected. This value will need to be tuned to suit your design. A larger value for RBIAS will result in a larger signal during BlinkUp™, but also makes your design more susceptible to saturation from ambient light. A smaller value reduces sensitivity to ambient light, but will produce a smaller signal during BlinkUp™. A 22-68kΩ resistor is a good place to start tuning.

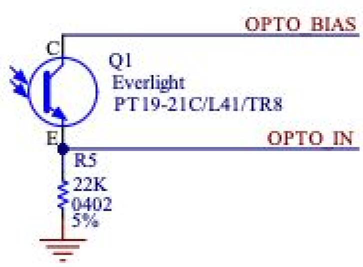


Fig. 6.1 — BlinkUp™ circuit

Pins connected to named imp006 IO

6.2. Bi-color (Red/Green) LED

This LED uses different color patterns to show different status conditions.

Your LED configuration must be able to produce three colors: red, green and amber. Most designs use a single bi-color LED to achieve this; some use two discrete LEDs and a lightpipe. Two-pin (“either-or”) bi-color LEDs are not supported: it must be possible to turn red and green on at the same time.

A 10kΩ resistor is required in parallel with the red LED to allow the imp006 to detect LED polarity. You may use either a common-cathode or a common-anode LED in your design; the imp006 will detect the LED polarity via the 10kΩ resistor.

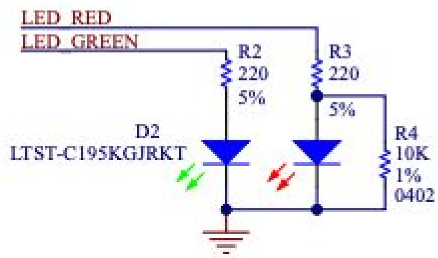


Fig. 6.2 — Bi-color LED circuit
Pins connected to named imp006 IO

6.2.1 LED Error Codes

The imp006 initially contains a secure bootstrap which, at first power on, will locate a correctly encrypted and signed image on the external QSPI flash. This image is then used to load impOS™ into the internal MCU flash, and make the device functional. This process takes around 20 seconds, during which time the device will show solid green with two short breaks.

If the imp006 is not connected to QSPI, the firmware is missing or the firmware fails the bootstrap code’s cryptographic checks, this state will be signalled on the BlinkUp™ status LED. The pattern it presents is two seconds of solid red. After this the MCU reboots, so the effect will be a slow red blink.

6.3. 32Mbit SPI Flash

An external 64Mbit (8MB) up to 128Mbit (16MB) SPI flash part is mandatory. The address space above `0x400000` (4096KB) can be used by your application through the imp API's **Spiflash** class. Areas below this address will be erased and reprogrammed by the OS, so applications using pre-programmed SPI flash components must not use space below this address.

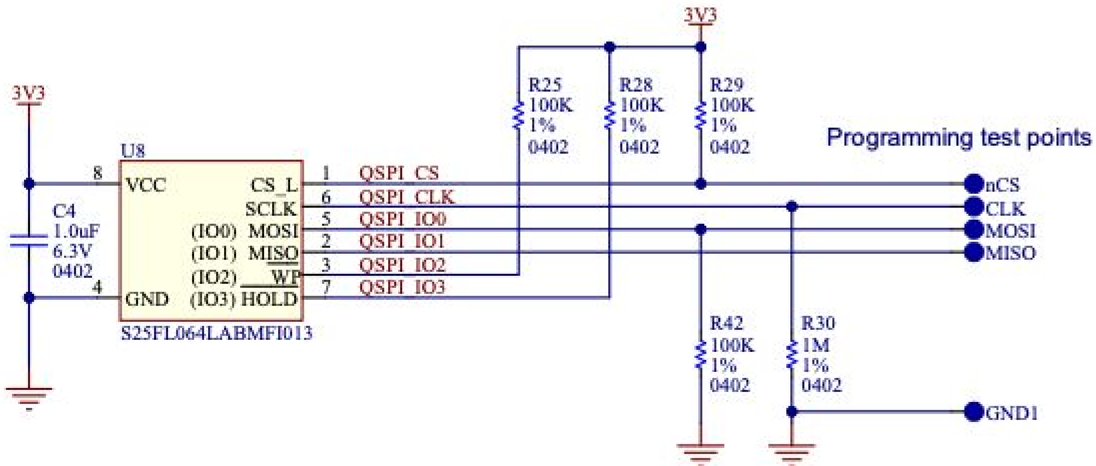


Fig. 6.3.1 — QSPI circuit

Pins connected to named imp006 IO

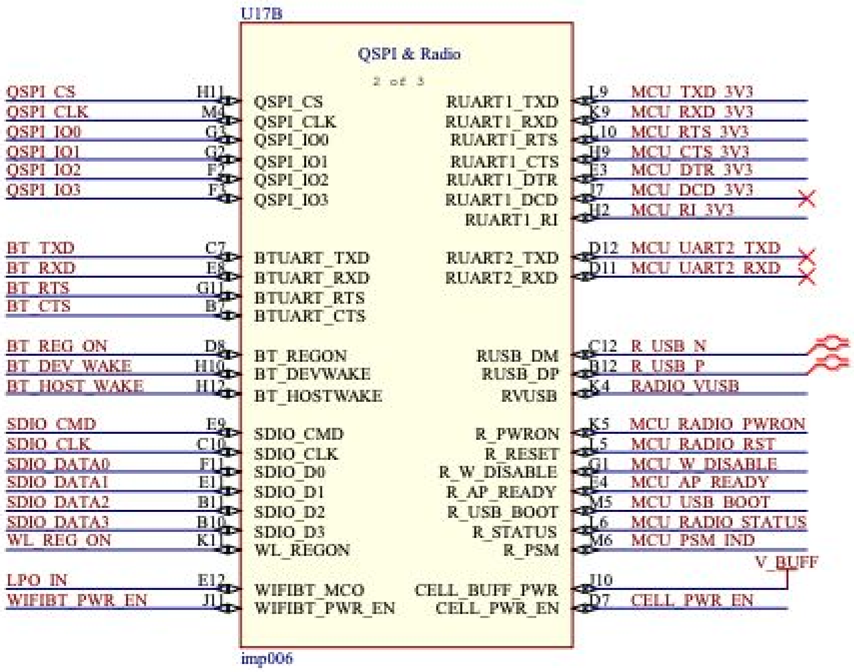


Fig. 6.3.2 — imp006 (partial) QSPI connections

The following SPI flash parts are directly supported:

Part Name	Chip ID	Notes
Macronix MX25R64xx	C2-28-17	Quad-read supported **Note** Old Macronix parts with tRES1 > 100μs may not work
Cypress S25FL064L	01-60-17	Quad-read supported
Winbond W25Q64JV	EF-40-17	Quad-read supported

Other SPI flashes should also work, although in 1-bit-wide mode only, provided they:

- Are 64Mbit (8MB) in size.
- Support 4KB, 32KB and 64KB erases (commands `0x20`, `0x52`, `0xD8`).
- Either support deep-power-down using commands `0xB9` (sleep) and `0xAB` (wake), or ignore those two commands.
- To reduce noise in the system, source termination resistors are recommended.

6.4. 32.768kHz Crystal

The imp006 does not include an internal timing crystal to run the internal Real-Time-Clock (RTC) while it is in deep sleep. If your design requires that the imp006 be capable of low-power deep sleep, it must include the crystal so that the imp can set timers and use them as a wake source. If no crystal is being used, you should ground the OSC32_IN pin.

A 32kHz crystal is also mandatory if your design is based on the imp006a variant as the 1MW module requires the 32kHz LPO clock to be driven.

Note Make sure the crystal has a load capacitance of 6pF and that suitably sized load capacitors are used. Higher capacitor values may prevent the oscillator from starting up. For more information, please see STMicro Appnote AN2867. To select a crystal, refer to Table 7, “Recommended crystal resonators for the LSE oscillator in STM32 MCUs”. Look for crystals that work with the F4_g2 series.

impOS™ configures the STM32 32kHz oscillator to high drive mode.

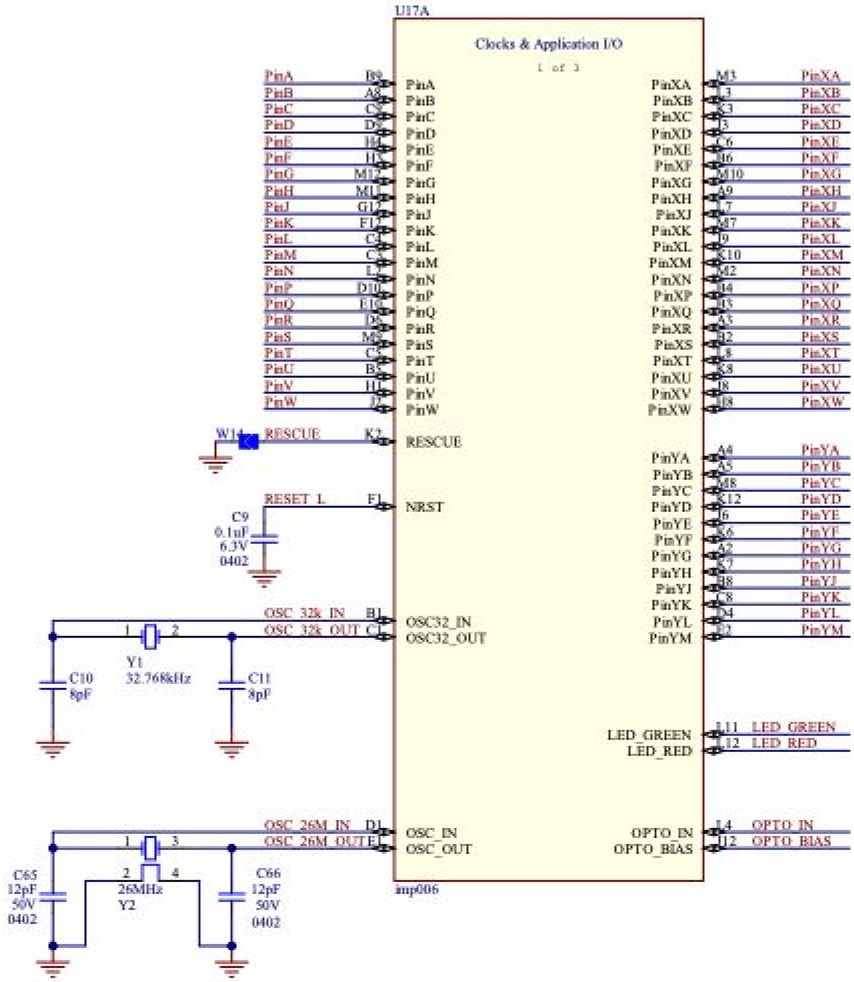


Fig. 6.4 — imp006 (partial) and clock circuits

6.5. 26MHz Crystal

The imp006 also requires a 26MHz crystal. This is the primary clock source for the STM32F423ZHJ6I MCU. This should be connected between the OSC_IN and OSC_OUT pins. This is used for all variants of the imp006.

Note 1 STMicro Appnote AN2867 has a list of suitable HSE crystals: see Table 6, “HSE oscillators embedded in STM32 MCUs / MPUs”.

Note 2 Ensure the crystal has suitably sized load capacitors for reliable start-up and accurate frequency keeping. Please consult the imp006 breakout layout for guidance on good component placement and signal routing.

7. Pin Layout

The following figure shows the imp006 pinout top view:

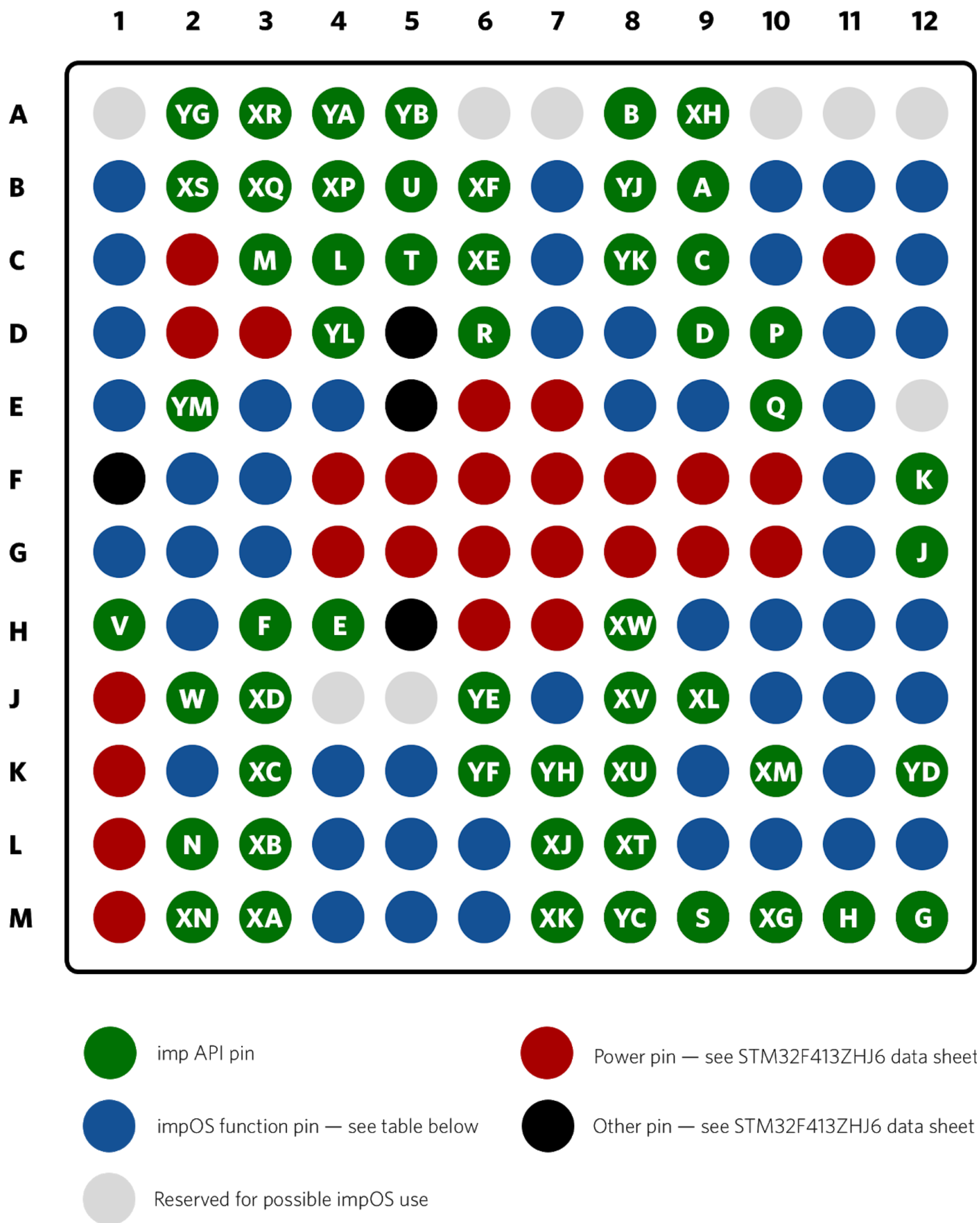


Fig. 7.1 — imp006 LGA pin functions

The following tables show how the STM32F423ZHJ6I's pins map to imp006 pins:

- GPIO/peripherals accessed via the imp API (green). For specific imp pin functions, please see the imp006 pin mux. Grid references are based on the co-ordinate system used in the diagram above.
- Specific imp-related functions (blue) which are not exposed to Squirrel. Please see the imp006 Breakout Board schematic for further details.
- STM32F423ZHJ6I GPIO pins (grey) that are reserved for possible impOS™ use.
- STM32F423ZHJ6I power pins (red).
- Other STM32F423ZHJ6I pins (black).

STM Grid Ref.	impOS Func.	STM Grid Ref.	impOS Func.	STM Grid Ref.	impOS Func.
A1	Reserved	B1	OSC32_IN	C1	OSC32_OUT
A2	pinYG	B2	pinXS	C2	VBAT
A3	pinXR	B3	pinXQ	C3	pinM
A4	pinYA	B4	pinXP	C4	pinL
A5	pinYB	B5	pinU	C5	pinT
A6	Reserved	B6	pinXF	C6	pinXE
A7	Reserved	B7	BTUART_CTS	C7	BTUART_TX
A8	pinB	B8	pinYJ	C8	pinYK
A9	pinXH	B9	pinA	C9	pinC
A10	Reserved	B10	SDIO	C10	SDIO
A11	Reserved	B11	SDIO	C11	VDDUSB
A12	Reserved	B12	USB_DATA +	C12	USB_DATA -

Table 7.1 — imp006 pins A1 through C12

STM Grid Ref.	impOS Func.	STM Grid Ref.	impOS Func.	STM Grid Ref.	impOS Func.
G1	RWDIS	H1	pinV	J1	VSSA
G2	QSPI	H2	RRING	J2	pinW
G3	QSPI	H3	pinF	J3	pinXD
G4	VSS	H4	pinE	J4	Reserved
G5	VDD	H5	BYPASS_REG	J5	Reserved
G6	VDD	H6	VSS	J6	pinYE
G7	VDD	H7	VCAP_1	J7	RDCD
G8	VSS	H8	pinXW	J8	pinXV
G9	VCAP_2	H9	RUART1_CTS	J9	pinXL
G10	VSS	H10	BTDW	J10	RV_BUFF
G11	BTUART_RTS	H11	QSPI	J11	PSU_EN
G12	pinJ	H12	BTHW	J12	OPTO_BIAS

Table 7.2 — imp006 pins G1 through J12

STM Grid Ref.	impOS Func.	STM Grid Ref.	impOS Func.	STM Grid Ref.	impOS Func.
K1	VREF —	L1	VREF +	M1	VDDA
K2	RESCUE	L2	pinN	M2	pinXN
K3	pinXC	L3	pinXB	M3	pinXA
K4	RVUSB	L4	OPTO_IN	M4	QSPI
K5	RPWRON	L5	RRST	M5	RUSBBT
K6	pinYF	L6	RSTATUS	M6	RPSM
K7	pinYH	L7	pinXJ	M7	pinXK
K8	pinXU	L8	pinXT	M8	pinYC
K9	RUART1_RX	L9	RUART1_TX	M9	pinS
K10	pinXM	L10	RUART1_RTS	M10	pinXG
K11	WLREGON	L11	RED LED	M11	pinH
K12	pinYD	L12	GREEN LED	M12	pinG

Table 7.3 — imp006 pins K1 through M12

7.1 Default Pin States

All pins default to tristate high impedance (floating).

7.2 The Rescue Pin

The imp006 pin RESCUE (PA1) may be used as an application rescue pin. If impOS™ has been set to Rescue Mode, RESCUE is sampled during a cold boot and its state determines whether Squirrel starts immediately (pin state high, the default) or after a ten-second period during which impOS™ attempts to connect to the impCloud™ and check for updated Squirrel (pin state low).

Rescue Mode is intended only for products which need Squirrel to run as quickly as possible when the device is connected to power: impOS™ will start Squirrel immediately upon a cold boot. Rescue Mode is not enabled by default and can only be enabled by calling **imp.enablerescuepin()** within Device Under Test (DUT) factory firmware.

When Rescue Mode is enabled, if RESCUE is externally driven low at cold boot, typically via an end-user activated push-button, then impOS™ will temporarily revert to the standard boot pattern: it will connect to the impCloud™ and check for updated Squirrel before it loads the Squirrel virtual machine.

RESCUE must be held high through a pull-up as its default, no-rescue state.

7.3 Resources

Related imp006 design resources, including the schematic symbol and PCB footprint, can be downloaded from the Electric Imp Dev Center.

8. The Electric Imp Platform

The imp006 is tightly bound to the Electric Imp Platform, a highly secure foundation for IoT applications.

The Electric Imp Platform starts with a device: the combination of the imp006 module, your choice of connectivity, your device-side application code and your peripheral sensors, actuators, displays, etc. Every device is twinned with an agent, a separate but always connected application hosted within the impCloud™. Each agent mediates and secures all of its device's communications with Internet-connected resources. Agents can make use of rich REST APIs to interact with web services and messaging protocols such as MQTT, and they themselves can process incoming HTTP requests. Customers can layer security on top of their agent-served APIs to suit their application.

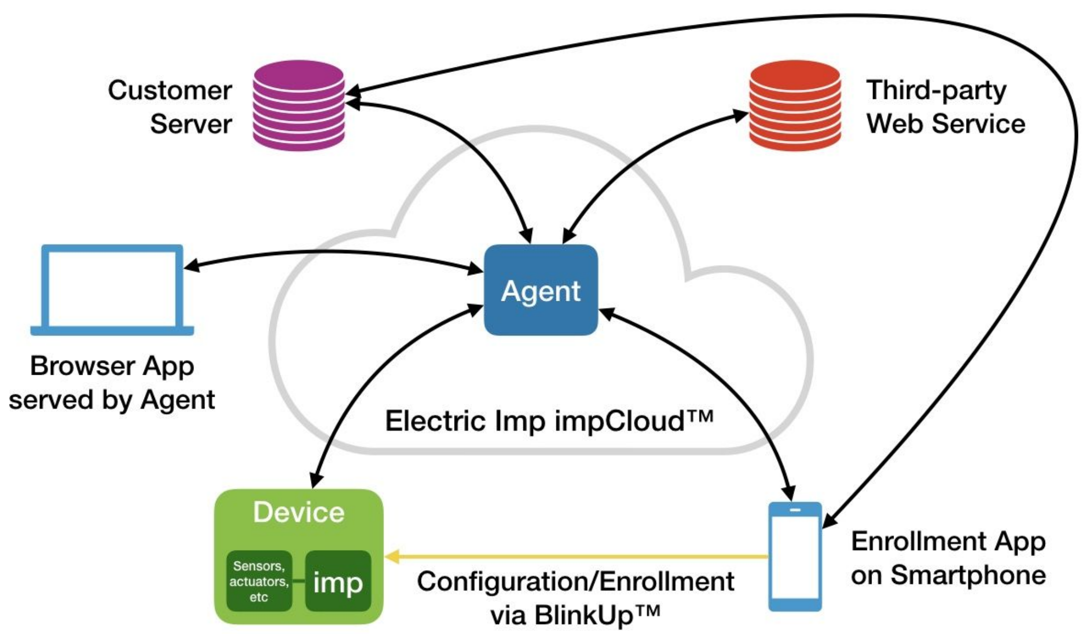


Fig. 8 — The Electric Imp Platform's key components and data flows

Together, agent code and device code form your Electric Imp application. Segment your application according to the strengths of each component: local hardware control and power management with the device code, Internet connectivity and data processing with the agent code. Both your agent code and your device code run securely in VMs hosted by, respectively, the cloud-side agent server and the device-side impOS™. These components are able to take advantage of our rich application foundation, the imp API. Their code can be created and managed within an online IDE or locally using command line tools. Updates are transmitted to groups of devices entirely over the air and ensure each device and its agent are always in sync.

9. Reference Designs

Electric Imp provides a series of reference designs based on the imp006 module. These can be produced as evaluation and early development devices in their own right, or can be used as the foundation for customers' own imp006-based hardware designs.

9.1 The imp006 Breakout Board

This board provides a full product evaluation, code development and product prototyping platform. It includes not only comprehensive power management options and a set of temperature, humidity and motion sensors, but also a user-controllable LED for visual feedback. The majority of the imp006's GPIO pins and peripherals are exposed on 0.1" headers. It provides cellular connectivity through a Quectel BG96 chip cellular modem and a Twilio Super SIM in MFF2 form. It also includes 802.11a/b/g/n/ac dual-band (2.4GHz and 5GHz) WiFi local-area networking and Bluetooth 5.0 personal-area networking thanks to the inclusion of a Murata LBEE5HY1MW wireless module.

9.2 The imp006 M.2 Board

This board provides customers with a means to incorporate a complete and removable imp sub-system in their products. The unit comprises an imp006 module with a Quectel BG96 CAT-M1/NB-IoT modem and a Twilio Super SIM in MFF2 form. The imp006's GPIO pins are routed through the M.2 (NGFF) connector. With all RF components and traces on the board, and a pre-approved radio module fitted, this provides an easy-to-use form-factor for customers who do not want to burden their PCBA designs with impedance control or fine-pitch BGA packages.

Note The imp006 M.2 Board is not a self-contained product. The board to which it will be connected must contain the circuitry required to deliver appropriate power supplies, optical BlinkUp™ and connect to the imp006 via the M.2 connector pins.

10. Further Design Resources

For further information about the imp006, please see the following resources provided by the Electric Imp Dev Center.

- imp006 design guide
- imp006 schematic symbol and PCB footprint
- imp006 pin mux listing
- imp006 BLE development guide
- imp006 GNSS development guide
- imp006 Breakout Board reference design
- imp006 M.2 reference design
- Get Started with the imp006 Breakout Board
- imp API documentation

For more detailed information on the imp006's STM32F423ZHJ6I MCU, please see the STMicro STM32F413x Datasheet.

For more detailed information on the Murata 1MW WiFi module, please see its datasheet.

For more detailed information on the Murata 1DX WiFi module, please see its datasheet.

To talk about how imp006 will enhance your IoT product, please contact **Twilio Sales**.

11. Document History

Date	Notes
February 16, 2021	Initial release
March 18, 2021	Updated with new STM MCU part number