Introduction to MC60 Development Board

25 March 2025

Supported Module List

Arduino uno R3

Arduino uno R3 SMD

Arduino Zero

Arduino Yun Rev2

Arduino UNO Wi-Fi Rev2

Arduino UNO R4 Wi-Fi

Arduino uno R4 Minima

Arduino Leonardo

Feature List

Module Features:

1- Quad-band: 850/900/1800/1900MHz

2- AT Commands: GSM 07.07, 07.05 and Enhanced AT Commands

3- TCP/UDP/HTTP/FTP/PPP

4- Jamming Detection

5- Audio

6- FOTA

7- Dual SIM, Single Standby

8- OpenCPU

9- GPS+GLONASS

10- QuecFastFix

MC60 Arduino Feature:

The MC60 header Development board is a compact and portable board designed specifically for Arduino.

The main equipped modules are the mc60 series wireless communication modules.

The Development board has a USB Type-C interface, making it convenient for development. Developers only need a USB Type-C cable to easily use the Development board.

Additionally, the Development board is compatible with the expansion board of Arduino, which can be directly used on the Development board.

Development Board Resources

Function Description

The main component and interface placement of the Development board is shown in the following figure:

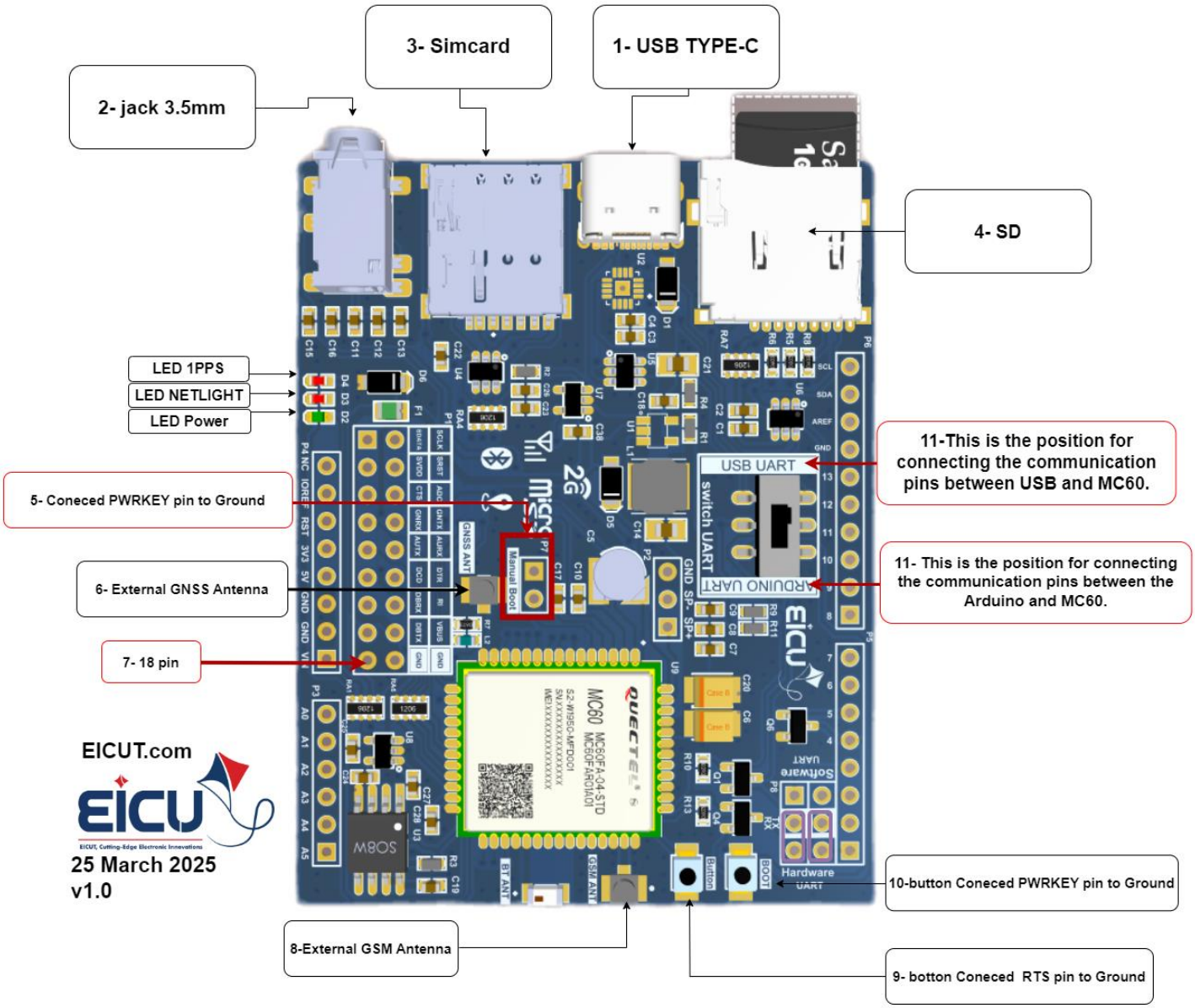
Development Board Configuration

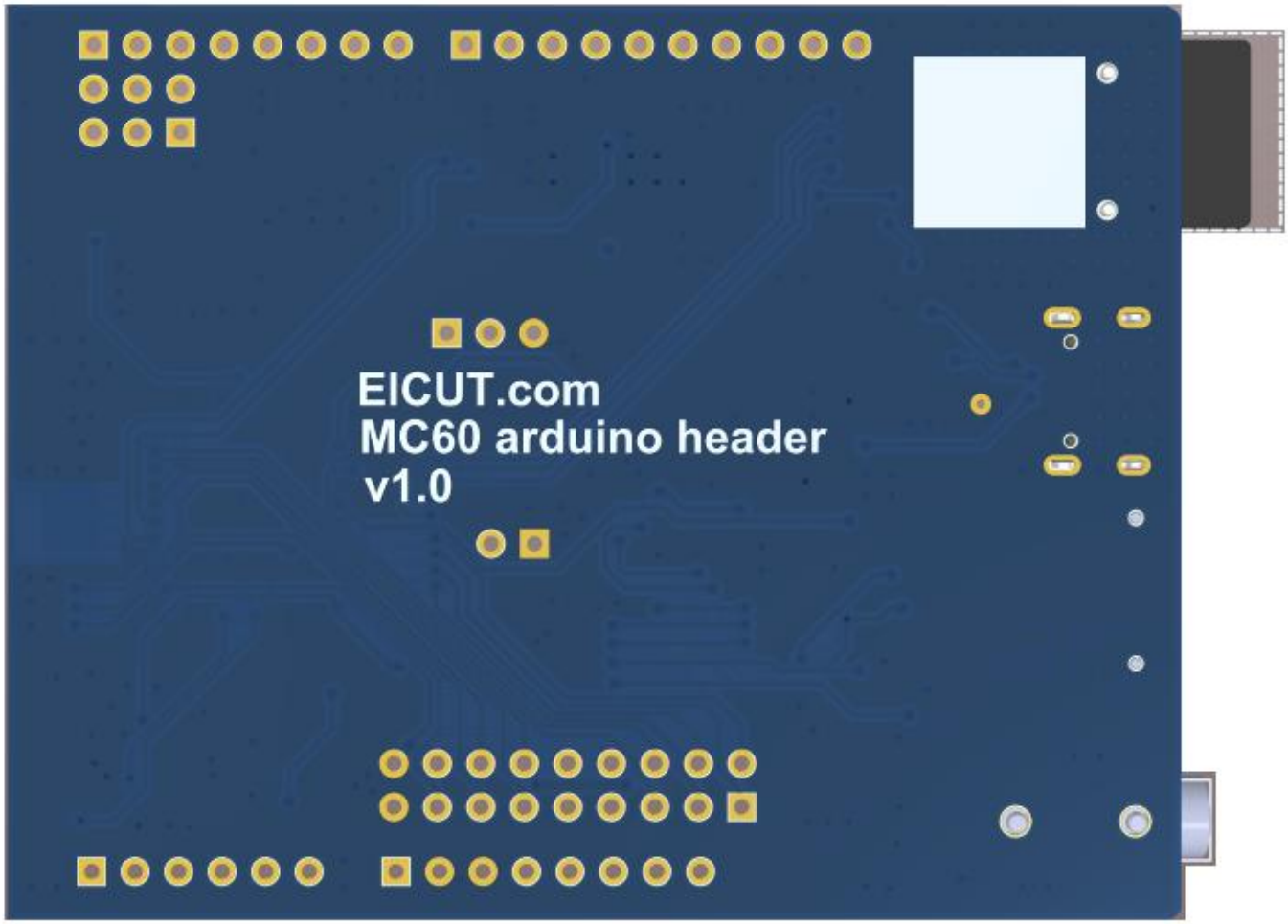
The detailed assignment of the peripheral interfaces on the Development board is as follows:

NO.	Name	Silkscreen	Comment
1	USB Type-C Interface	-	-
2	Audio jack 3.5 mm	-	-
3	Sim card case	-	4FF SIM Card 12.3* 8mm
4	SD card case	-	Micro SD Card
5	Manual boot	Manual Boot	Push button
6	External GNSS Antenna	GNSS ANT	UFL ANTENNA
7	18pin	-	header 2.54mm
8	External GSM Antenna	GSM ANT	UFL ANTENNA
9	button	button	If push Connect RST pin to ground
10	boot	BOOT	If push Connect PWRKEY pin to ground and turn on/off MC60
11	UART Switch	UART Arduino/USB	If the switch is on the Arduino UART side, communication between the Arduino and the MC60 will be established and if switch is on the USB UART side, communication between the Arduino and the MC60 will be established

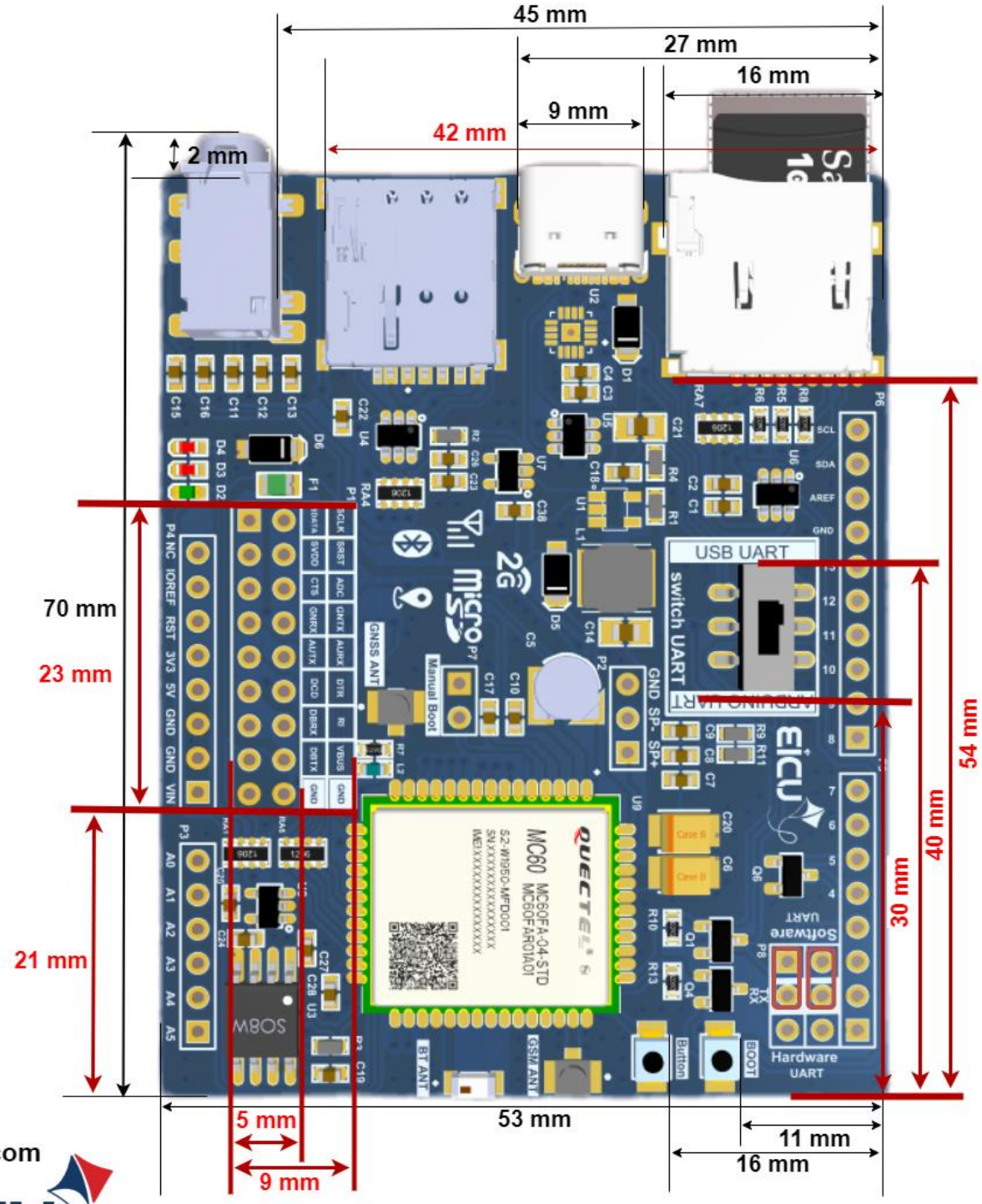
The Development board has 3 functional indication LEDs, as follows:

- **D4**: LED Connect 1PPS
- **D3**: Power indication LED
- **D2**: LED Connect NETLIGHT

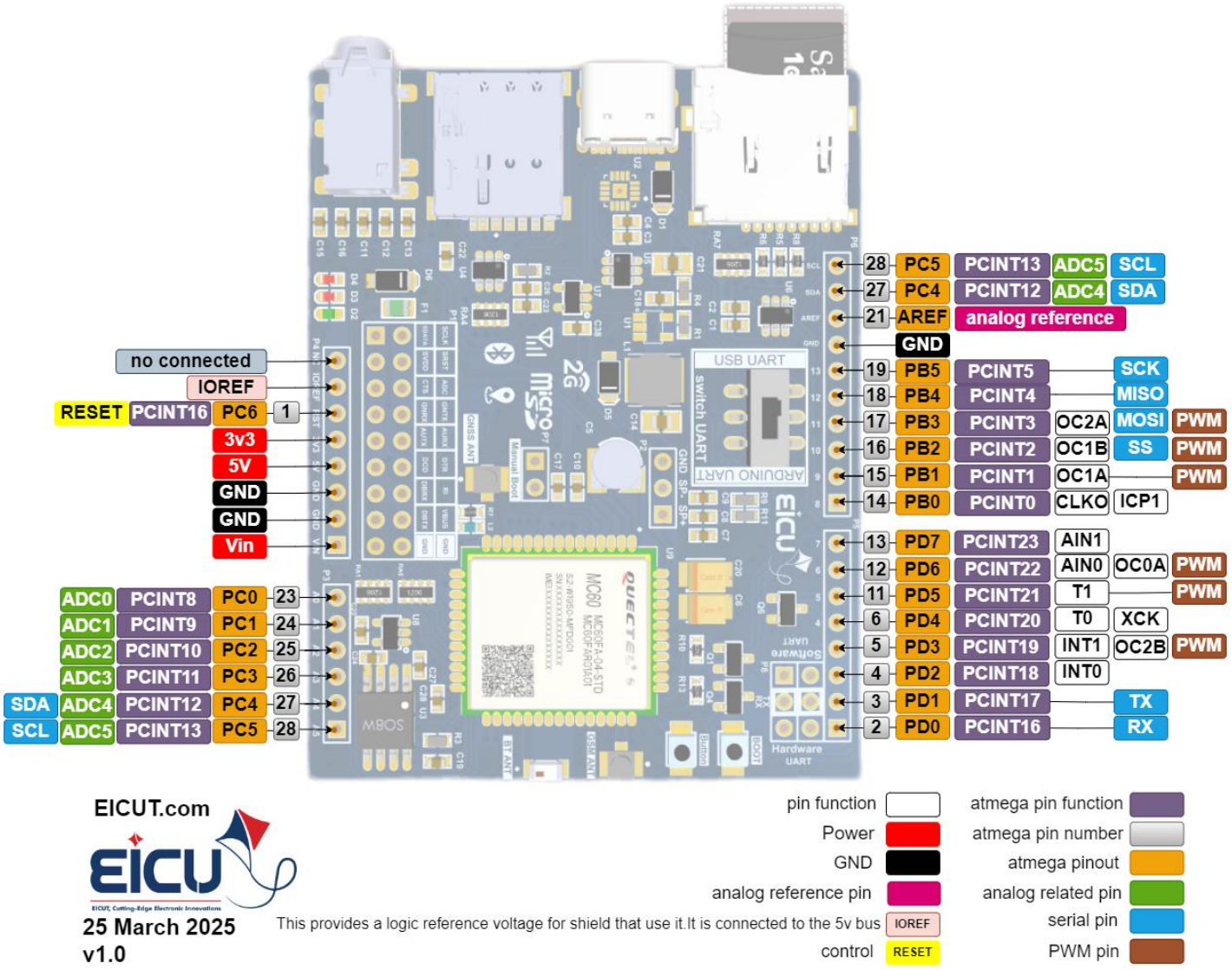


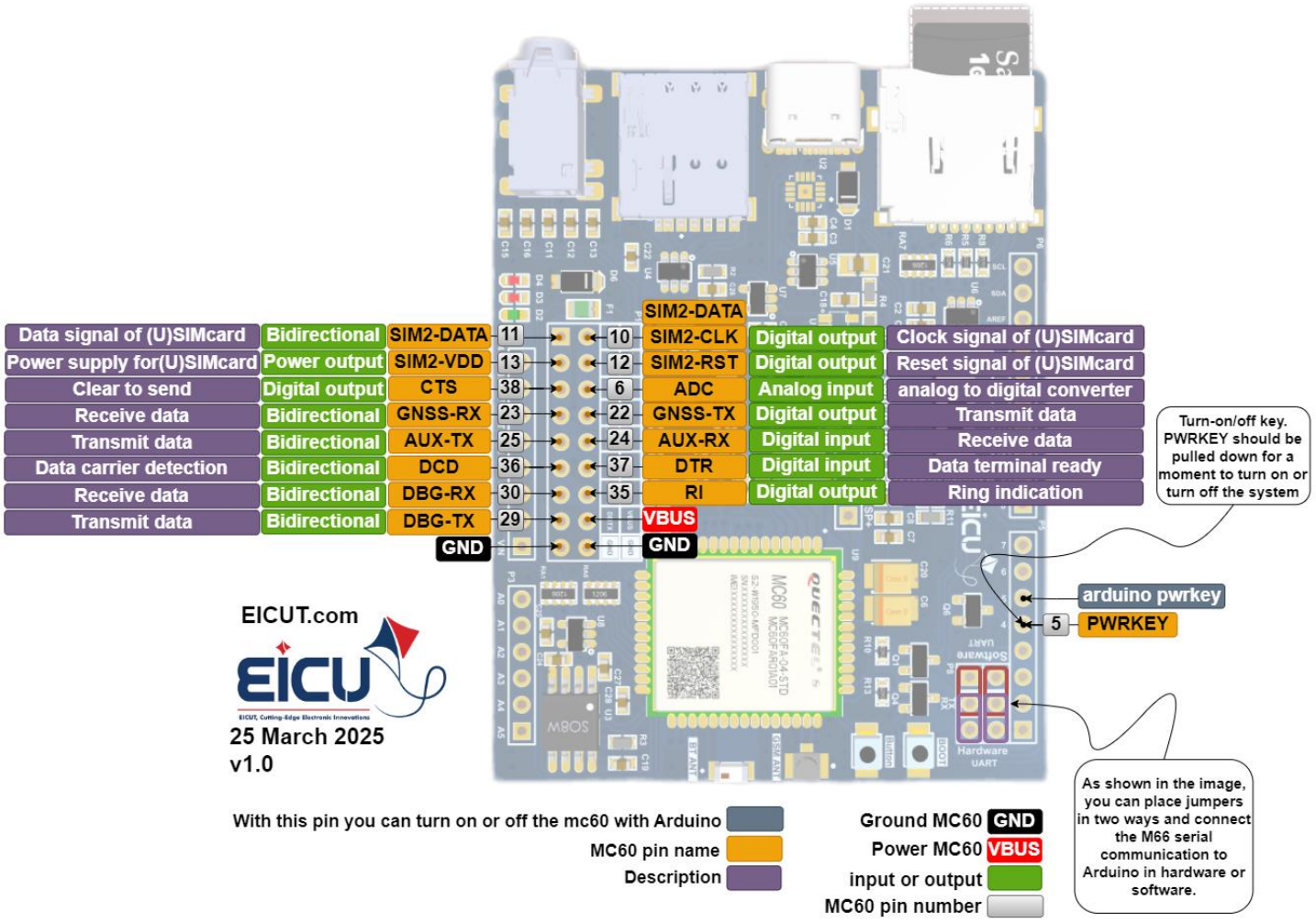

EICUT.com
MC60 arduino header
v1.0

- Development board dimensions:



The main pin placement of the Development board is shown in the following figure:





I tip

As shown in the image above, Arduino can communicate with the module in two ways: software UART and hardware UART.

Set the switch to Arduino UART mode and connect with a jumper depending on your needs.

I Development Board Interfaces

- **Power header**

NO.	Name	Silkscreen	Comment
8-Pin	NC	NC	-
8-Pin	IOREF	IOREF	This provides a logic reference voltage for shield that use it. It is connected to the 5v bus
8-Pin	RST	RST	Reset Arduino
8-Pin	3v3	3v3	3.3 V Arduino regulator don't Connect to power the module
8-Pin	5v	5v	5 V Arduino regulator Do not power the module from the Arduino, but the module can power the Arduino.
8-Pin	GND	GND	Ground
8-Pin	GND	GND	Ground
8-Pin	Vin	Vin	This pin is used to power the Arduino Uno board using an external power source. The voltage should be within the range mentioned above.

- **ADC header**

NO.	Name	Silkscreen	Comment
6-Pin	ADC0	A0	The Arduino Uno has 6 analog pins, which utilize ADC (Analog to Digital converter). These pins serve as analog inputs but can also function as digital inputs or digital outputs.
6-Pin	ADC1	A1	
6-Pin	ADC2	A2	
6-Pin	ADC3	A3	
6-Pin	ADC4	A4	
6-Pin	ADC5	A5	

Analog to Digital Conversion

ADC stands for Analog to Digital Converter. ADC is an electronic circuit used to convert analog signals into digital signals. This digital representation of analog signals allows the processor (which is a digital device) to measure the analog signal and use it through its operation.

Arduino Pins A0-A5 are capable of reading analog voltages. On Arduino the ADC has 10-bit resolution, meaning it can represent analog voltage by 1,024 digital levels. The ADC converts voltage into bits which the microprocessor can understand.

One common example of an ADC is Voice over IP (VoIP). Every smartphone has a microphone that converts sound waves (voice) into analog voltage. This goes through the device's ADC, gets converted into digital data, which is transmitted to the receiving side over the internet.

- **Digital header**

NO.	Name	kscreen	Comment
18-Pin	PD0/TX	0	Serial Communication
18-Pin	PD1/RX	1	
18-Pin	PD2	2	
18-Pin	PD3/PWM	3	
18-Pin	PD4	4	
18-Pin	PD5/PWM	5	
18-Pin	PD6/PWM	6	
18-Pin	PD7	7	
18-Pin	PB0	8	
18-Pin	PB1/PWM	9	
18-Pin	PB2/PWM/SS	10	SPI communication
18-Pin	PB3/PWM/MOSI	11	
18-Pin	PB4/MISO	12	
18-Pin	PB5/SCK	13	
18-Pin	Ground	GND	-
18-Pin	AREF	AREF	
18-Pin	PC4/SDA	SDA	I2C communication
18-Pin	PC5/SCL	SCL	

- **MC60 header 2*9**

NO.	Name	Silkscreen	Comment
2*9-Pin	Sim2-DATA	SDATA	You can use it as an output and input.
2*9-Pin	Sim-CLK	SCLK	Digital output
2*9-Pin	Sim2-VDD	SVDD	Power output
2*9-Pin	Sim2-RST	SRST	Digital output
2*9-Pin	Clear to send	CTS	Digital output
2*9-Pin	Analog input	ADC	Analog input
2*9-Pin	GNSS-RX	GNRX	You can connect the GNSS-RX to AUX-TX and GNSS-TX to AUX-RX by jumpers for internal communication between the GPS module and itself.
2*9-Pin	GNSS-TX	GNTX	
2*9-Pin	AUX-TX	AUTX	
2*9-Pin	AUX-RX	AURX	
2*9-Pin	Data carrier detection	DCD	You can use it as an output and input.
2*9-Pin	Data terminal ready	DTR	Digital output
2*9-Pin	DBG-RX	DBRX	You can use it as an output and input.
2*9-Pin	Ring indication	RI	Digital output
2*9-Pin	DBG-TX	DBTX	You can use it as an output and input.
2*9-Pin	VBUS	VBUS	This pin is powered by USB.
2*9-Pin	Ground	GND	-
2*9-Pin	Ground	GND	-

What does digital mean?

Digital is a way of representing voltage in 1 bit: either 0 or 1. Digital pins on the Arduino are pins designed to be configured as inputs or outputs according to the needs of the user. Digital pins are either on or off. When ON they are in a HIGH voltage state of 5V and when OFF they are in a LOW voltage state of 0V.

On the Arduino, When the digital pins are configured as output, they are set to 0 or 5 volts.

When the digital pins are configured as input, the voltage is supplied from an external device. This voltage can vary between 0-5 volts which is converted into digital representation (0 or 1). To determine this, there are 2 thresholds:

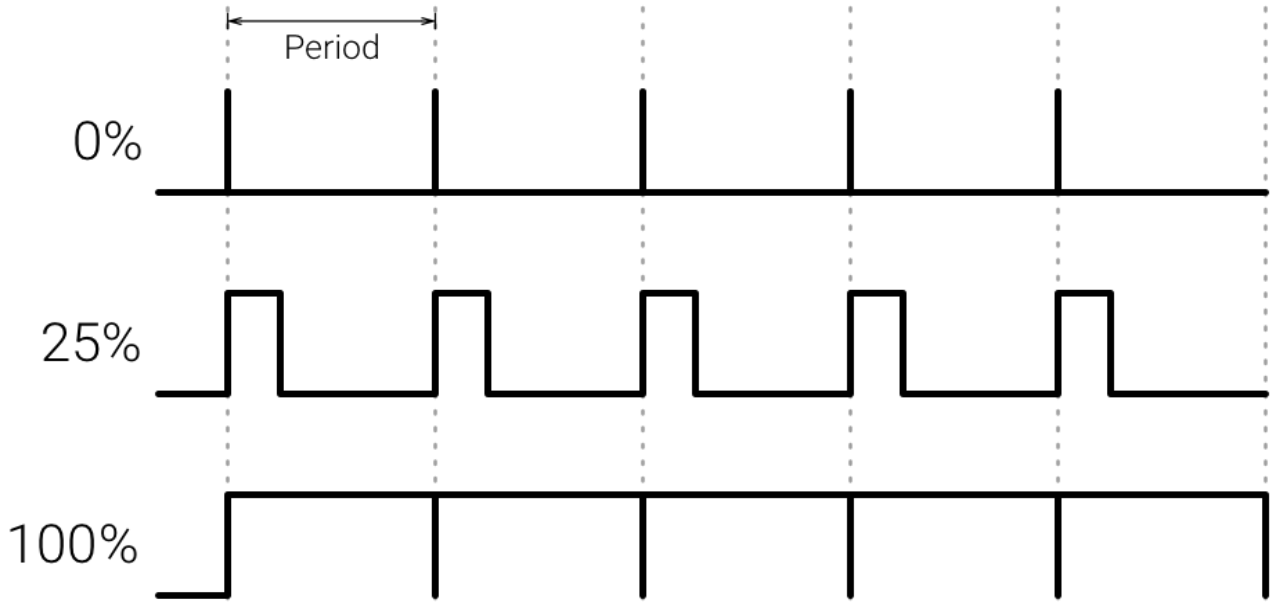
- Below 0.8v - considered as 0.
- Above 2v - considered as 1.

When connecting a component to a digital pin, make sure that the logic levels match. If the voltage is in between the thresholds, the returning value will be undefined.

What is PWM?

In general, Pulse Width Modulation (PWM) is a modulation technique used to encode a message into a pulsing signal. A PWM is comprised of two key components: frequency and duty cycle. The PWM frequency dictates how long it takes to complete a single cycle (period) and how quickly the signal fluctuates from high to low. The duty cycle determines how long a signal stays high out of the total period. Duty cycle is represented in percentage.

In Arduino, the PWM enabled pins produce a constant frequency of ~ 500Hz, while the duty cycle changes according to the parameters set by the user. See the following illustration:



PWM signals are used for speed control of DC motors, dimming LEDs and more.

Communication Protocols

Serial (TTL) - Digital pins 0 and 1 are the serial pins of the Arduino Uno.

They are used by the onboard USB module.

What is Serial Communication?

Serial communication is used to exchange data between the Arduino board and another serial device such as computers, displays, sensors and more. Each Arduino board has at least one serial port. Serial communication occurs on digital pins 0 (RX) and 1 (TX) as well as via USB. Arduino supports serial communication through digital pins with the Software Serial Library as well. This allows the user to connect multiple serial-enabled devices and leave the main serial port available for the USB.

Software serial and hardware serial - Most microcontrollers have hardware designed to communicate with other serial devices. Software serial ports use a pin-change interrupt system to communicate. There is a built-in library for Software Serial communication. Software serial is used by the processor to simulate extra serial ports. The only drawback with software serial is that it requires more processing and cannot support the same high speeds as hardware serial.

SPI - SS/SCK/MISO/MOSI pins are the dedicated pins for SPI communication. They can be found on digital pins 10-13 of the Arduino Uno and on the ICSP headers.

What is SPI?

Serial Peripheral Interface (SPI) is a serial data protocol used by microcontrollers to communicate with one or more external devices in a bus like connection. The SPI can also be used to connect 2 microcontrollers. On the SPI bus, there is always one device that is denoted as a Master device and all the rest as Slaves. In most cases, the microcontroller is the Master device. The SS (Slave Select) pin determines which device the Master is currently communicating with.

SPI enabled devices always have the following pins:

MISO (Master in Slave out) - A line for sending data to the Master device

MOSI (Master out Slave In) - The Master line for sending data to peripheral devices

SCK (Serial Clock) - A clock signal generated by the Master device to synchronize data transmission.

I2C - SCL/SDA pins are the dedicated pins for I2C communication. On the Arduino Uno they are found on Analog pins A4 and A5.

What is I2C?

I2C is a communication protocol commonly referred to as the “I2C bus”. The I2C protocol was designed to enable communication between components on a single circuit board. With I2C there are 2 wires referred to as SCL and SDA.

SCL is the clock line which is designed to synchronize data transfers.

SDA is the line used to transmit data.

Each device on the I2C bus has a unique address, up to 255 devices can be connected on the same bus.

Aref - Reference voltage for the analog inputs.

Interrupt - INT0 and INT1. Arduino Uno has two external interrupt pins.

External Interrupt - An external interrupt is a system interrupt that occurs when outside interference is present. Interference can come from the user or other hardware devices in the network. Common uses for these interrupts in Arduino are reading the frequency a square wave generated by encoders or waking up the processor upon an external event.

Arduino has two forms of interrupt: External Pin Change

There are two external interrupt pins on the ATmega168/328 called INT0 and INT1. both INT0 and INT1 are mapped to pins 2 and 3. In contrast, Pin Change interrupts can be activated on any of the pins

Getting Started Preparation

Step 1: Connect the Development Board

Connect the Development board to the computer's USB port using a USB Type-C cable for power supply.

Step 2: Turn on the Development board

Long press the PWRKEY button to turn on the board. Before pressing the PWRKEY button, make sure the power supply voltage is stable. It is recommended to have a time interval of not less than 30 ms between powering up and pressing the PWRKEY button. If you want the board to power up and turn on automatically without the need for a turn-off function, you can directly short the two pins with PWK_AUTO silkscreen on the 2-Pin (Manual Boot) header.

Step 3: Install the 343p IC driver from the link below.

https://www.wch-ic.com/products/CH343.html

Step 4: Don't forget that when you want to connect the module to the computer, the switch mode is on USB UART.

Step 5: In the next step, install the Qnavigator software from the link below.

https://www.quectel.com/download/qnavigator_v1-5/

Step 6: After installing the software, follow the software instructions and select your board in the port section, set the baud rate to 115200, and click Connect.

Resource Download