

IamaPLC Communication: I²C

I²C (*Inter-Integrated Circuit*; pronounced as “eye-squared-C”) is also known as I2C, I²C, or IIC. It is a synchronous, multi-master/multi-slave (controller/target), packet-switched, single-ended, serial communication bus invented in 1982 by Philips Semiconductors.



I²C is a widely used digital communication standard in many devices. The protocol employs a two-wire interface, enabling multiple controllers and peripheral devices. This application note highlights key aspects of the protocol as a guide to using I²C for communicating with controller devices.

This note covers both the protocol and the physical layer for I²C communication. Since I²C is often used with data converter devices, examples include communication with a DAC and an ADC for reading and writing registers.

Several companies, such as Siemens, NEC, Texas Instruments, STMicroelectronics, Motorola, Nordic Semiconductor, and Intersil, have released compatible I²C products since the mid-1990s.

Protocols Similar to I²C

The I²C specification covers several other communication protocols based on I²C. These protocols may be similar and compatible with I²C communication and can be used for specific applications. Some protocols also include defined commands and application-specific extensions for their systems. This section briefly describes the applications for these protocols, but their systems, applications, and uses are detailed elsewhere.

The first of these similar protocols is the Two-Wire Interface (TWI), which is equivalent to I²C.

However, there are some minor differences. TWI does not support a START byte and does not support high-speed modes. Generally, TWI-compatible devices are expected to be compatible with I²C, and the protocol can be seen with the same logic analyzers.

The System Management Bus, or SMBus[®], is a protocol similar to I²C that is tailored to a specific function. It is commonly used in servers and computer motherboards for power source management. The protocol is very similar to I²C in the communication protocol and can be understood by an I²C controller.

The SMBus protocol has additional features compared to I²C. The SMBus can dynamically set addresses, allowing for quick communications at the start-up of a system. Also, the bus has a 35-ms timeout, which prevents one device from indefinitely tying up the bus. The protocol also has a packet error checking for error detection in data communication. There is an additional line called SMBAlert that is used by target devices as an interrupt to tell the controller about certain events detected by the target device.

The Power Management bus, or PMBus[®], is a variant of SMBus defined by Intel. It is used in the digital management of power supplies. This protocol also defines specific commands to retrieve data

about the system's voltage, current, and power.

Intelligent Platform Management Interface IPMI is another I²C-based protocol. This protocol is used by baseboard management controllers (BMCs) to monitor and manage the system's CPU, firmware, and operating system in autonomous computer subsystems. The protocol utilizes a standardized message-based interface for computer motherboards or servers. The BMC is always running even when the central system is off, allowing for operation, measurement, and remote management of a system.

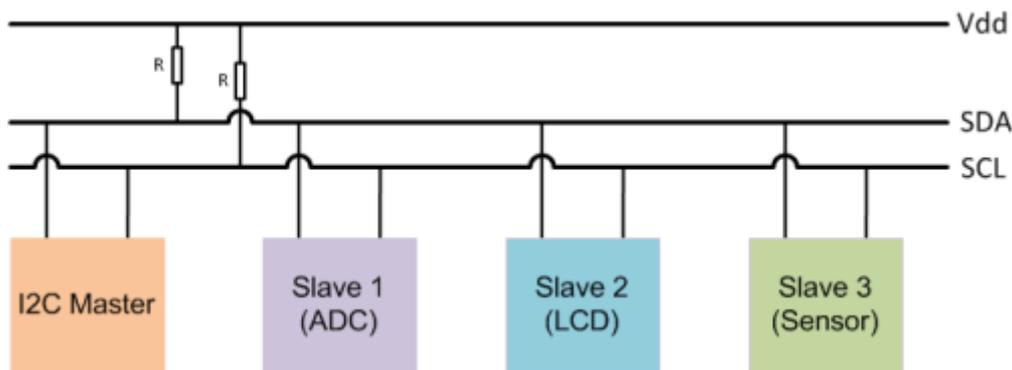
Several other similar protocols are discussed in the I²C specifications. The Advanced Telecommunications Computing Architecture (ATCA) is a follow-on to CompactPCI, used in rack-mounted telecom hardware.

Display Data Channel (DDC) is a protocol that monitors or displays information, allowing hosts to control display functions. Finally, C-Bus is another protocol derived from I²C. As mentioned in the reserved address section, this protocol is used in some parts of the world for home and building automation.

I²C is similar in concept to [1-Wire](#).

I²C specifications

- In the most common case, the I²C bus contains only one master, but the possibility of a multimaster is also not excluded
- I²C uses an open-ended pair of wires for data transfer: the serial data line (SDA) and the serial clock signal (SCL). The two wires must be pulled to the operating voltage with resistors. The cables typically operate with a voltage of 5V or 3.3V, but a different voltage specification is also not excluded.



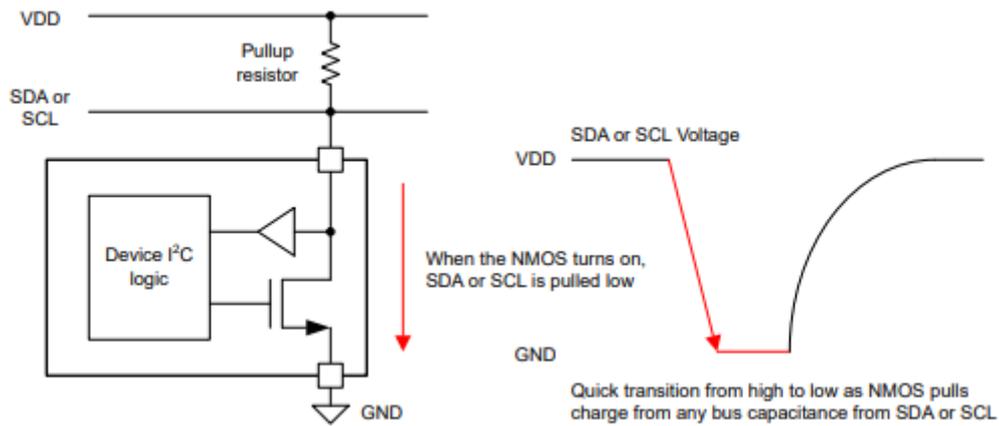
- Pull-up resistors are almost always 4.7k
 - In principle, the maximum distance of the bus is 7.6 m, but it is typically used within 2-3 m.
 - The bus typically uses 7- or 10-bit addressing, but sometimes the 16-bit solution also occurs.
- Typical transmission speeds:

Mode	Speed
Low speed mode	10 kbit / sec
standard mode	100 kbit / sec
fast mode (FM)	400 kbit / sec
fast mode plus (FM+)	1 Mbit / sec

Mode	Speed
high-speed mode (HS)	1.7 .. 3.4 Mbit / sec
Ultra-fast mode (UFm)	5 Mbit / sec

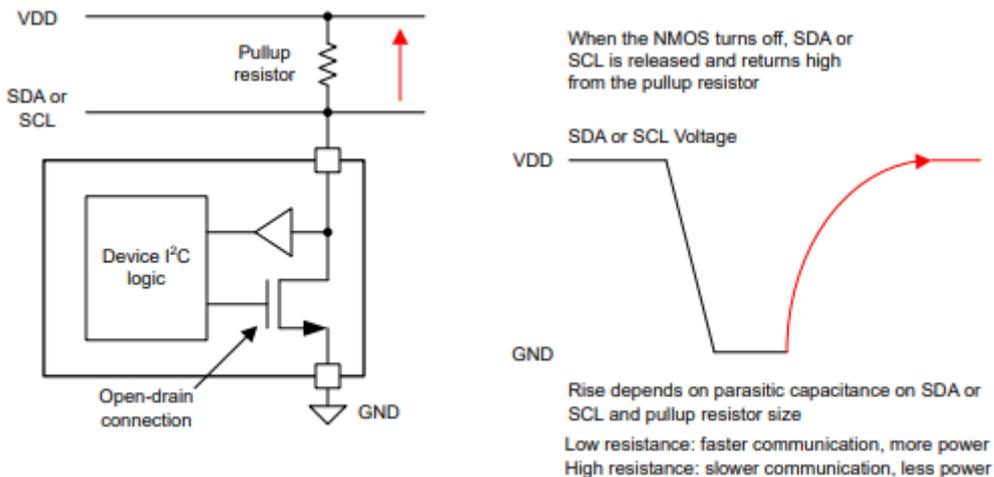
Open-Drain Connection

The open-drain connections are used on both SDA and SCL lines and connect to an NMOS transistor. This open-drain connection controls the I²C communication line, pulling the line low or releasing it high. The open-drain refers to the NMOS bus connection when the NMOS is turned OFF. The next pic shows the open-drain connection as the NMOS is turned on:



The NMOS is set on or off to set the voltage level of the SDA or SCL line. When the NMOS is on, the device pulls current through the resistor to ground, pulling the open-drain line low. Typically, the transition from high to low for I²C is as fast as the NMOS pulls down on SDA or SCL. The NMOS drive strength and any bus capacitance on SDA or SCL determine the transition speed.

When the NMOS turns off, the device stops pulling current, and the pull-up resistor pulls the SDA or SCL line to VDD. The following picture shows an open-drain line as the NMOS is turned off. The pull-up resistor pulls the line high. The transition of the open-drain line is slower because the line is pulled up against the bus capacitance and is not actively driven.



I²C addresses

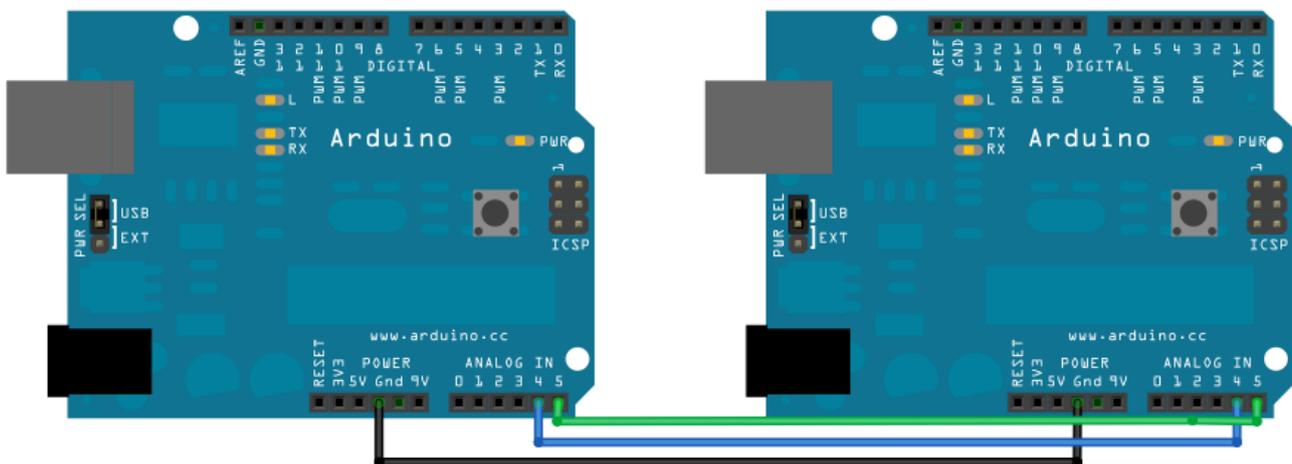
Different manufacturers' applications and addresses with different lengths (7, 8, or 10 bits) significantly limit the usable address range, which is practically **0x08 .. 0x77**.

I²C Arduino

The I²C protocol utilizes two lines to send and receive data: a serial clock pin (SCL), which the Arduino Controller board pulses regularly, and a serial data pin (SDA) over which data is exchanged between the two devices.

In I²C, one controller device has one or more peripheral devices connected to the controller's SCL and SDA lines.

As the clock line changes from low to high (known as the rising edge of the clock pulse), a single bit of information is transferred from the board to the I²C device over the SDA line. As the clock line continues to pulse, more and more bits are sent until a sequence of 7 or 8 bits, along with a command or data, is formed. When this information is sent, bit by bit, the called-upon device executes the request and transmits its data back, if required, to the board over the same line using the clock signal still generated by the Controller on SCL as timing.



Each device on the I²C bus is functionally independent of the controller, but it will respond with information when prompted by the controller.

Because the I²C protocol allows each enabled device to have its own unique address, and both controller and peripheral devices take turns communicating over a single line, your Arduino board can communicate (in turn) with many devices or other boards, using just two pins of your microcontroller.

- The controller sends out instructions through the I²C bus on the data pin (SDA), and the instructions are prefaced with the address, so that only the correct device listens.
- Then there is a bit signifying whether the controller wants to read or write.
- Every message needs to be acknowledged to combat unexpected results. Once the receiver has accepted the previous information, it notifies the controller, allowing it to proceed to the next

set of bits.

- 8 bits of data
- Another acknowledgement bit
- 8 bits of data
- Another acknowledgement bit

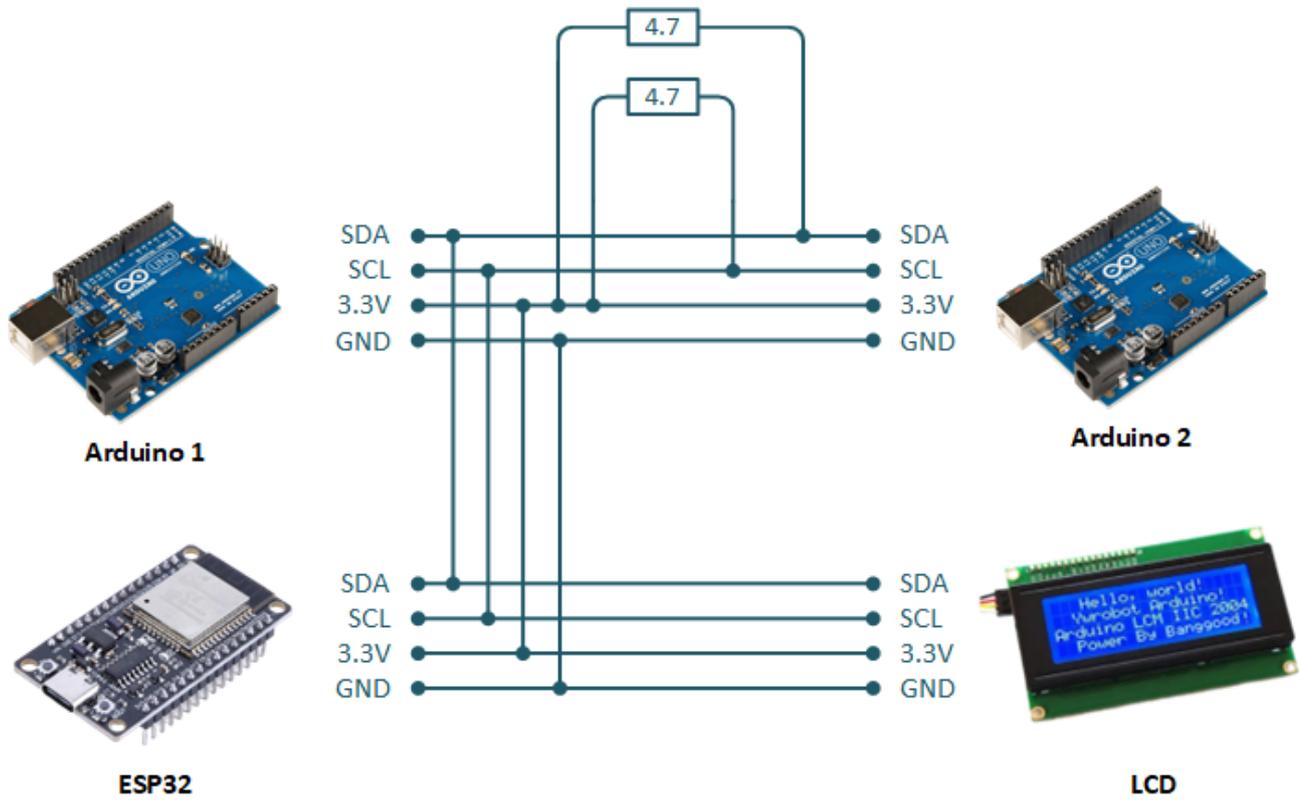
More from [Arduino I²C protocol](#)

Arduino applications with I²C communication

- [Arduino LCD display](#)
- [INA 226: 36V, 16-bit, ultra-precise I²C output current/voltage/power monitor](#)
- [Adafruit AM2315C Temperature/Humidity sensor](#)
- [Bosch BMP 180 Temperature/Air Pressure sensor](#)
- [Bosch BME 280 Temperature/Humidity/Air Pressure sensor](#)
- [Bosch BME 680 Temperature/Humidity/Air Pressure sensor](#)
- [Sensirion CCS811 CO²/VOC sensor](#)
- [Omron D6T series Non-Contact Thermal Sensor](#)
- [DHT 20 Temperature/Humidity sensor](#)
- [ScioSense ENS-160 TVOC/eCO₂ sensor](#)
- [Texas Instruments HDC 1080 Temperature/Humidity sensor](#)
- [TE Connectivity HTU 21 Temperature/Humidity sensor](#)
- [Sensirion SGP/SHT Series Temperature/Humidity sensors](#)

I²C Arduino with 3.3V

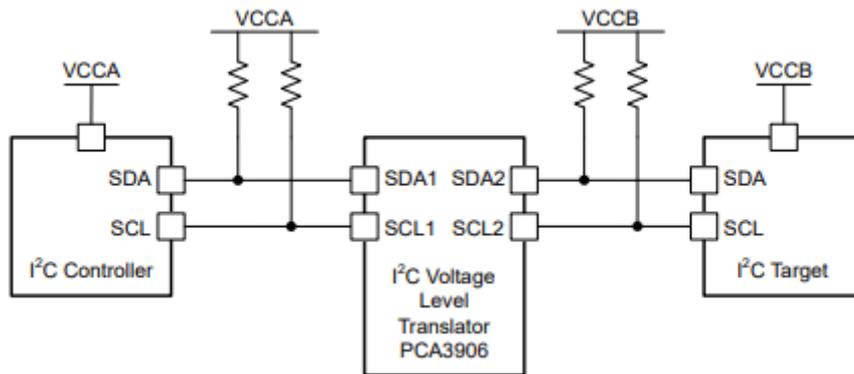
The I²C communication primarily operates at 5V, but in certain cases, such as when a 3.3V base voltage unit (ESP32) is present in the network, 3.3V is also sufficient for the 5V units. The voltage level can be stabilized with 4.7 kOhm pull-up resistors (SDO, SCL):



lamaConception I²C communication

lamaPlc.com

According to Texas Instruments' analysis, the above solution does not work because the high voltage levels do not match between 5V and 3.3V. In the case of 5V, 3.5V is the high-level threshold, as seen in [here](#).



They recommend using the [PCA9306 IC](#) for this case.

Sources

- [Wikipedia I²C](#)
- [Texas Instruments: A Basic Guide to I2C](#)
- [Texas Instruments: Understanding the I2C Bus](#)

I²C topics on lamaPLC

Page	Date	Tags
• lamaPLC Communication: 1-Wire	2025/05/31 21:56	1-wire , communication , bus , microlan , i2c , uart , usart , ds18b20
• lamaPLC Communication: I²C	2025/09/23 19:25	i2c , i c , smbus , philips , bus , communication , arduino
• LamaPLC: AHT20 / BMP280 Modul	2026/02/15 20:33	bmp280 , aht20 , adafruit , temperature , humidity , pressure , sensor , arduino , code , i2c
• LamaPLC: APDS - Avago ALS and proximity detection sensors with I²C communication	2026/02/14 22:24	avago , apds-9900 , apds-9930 , apds-9960 , als , proximity , detection , gesture recognition , gesture , i2c , communication , sensor , arduino , code
• LamaPLC: BMP/BME Bosch Temperature/Humidity/Pressure sensors with I²C communication	2026/02/15 20:40	bme280 , bme680 , bmp180 , bmp280 , bosch , temperature , humidity , pressure , sensor , arduino , i2c , communication , cjmcu
• LamaPLC: CJMCU-219/INA-219 breakout board/IC with I²C communication	2026/02/14 23:37	cjmcu-219 , ina-219 , ina219 , breakout board , i2c , communication , sensor , voltage , current , arduino , code , cjmcu
• LamaPLC: CJMCU-3216 / AP-3216 integrated digital ambient light and proximity sensor module/IC with I²C communication	2026/02/14 22:40	cjmcu-3216 , cjmcu , ap-3216 , ap3216 , ambient light , proximity , sensor , arduino , code , i2c , communication
• LamaPLC: D6T Omron Non-Contact Thermal Sensors with I²C communication	2026/02/14 18:19	d6t , d6t-32l , d6t-44l , d6t-8l , d6t-1a , omron , non-contact , thermal , sensor , i2c , arduino , code
• LamaPLC: DPS Infineon Temperature/Pressure sensors with I²C communication	2026/02/14 18:11	dps310 , infineon , temperature , pressure , sensor , arduino , i2c , communication , code
• lamaPLC: Energy, power, current, and voltage	2025/05/31 21:32	i2c , i c , communication , arduino , energy , power , current , sensor , ina226
• LamaPLC: ENS ScioSense Multi-gas sensors with I²C communication	2026/02/14 19:29	ens160 , sciosense , gas-quality , i2c , communication , sensor , arduino , code , eco2 , tvoc , aqi , indoor air quality , iaq , co2 , voc
• lamaPLC: ESP32 / ESP8266	2025/11/21 23:07	esp8266 , esp32 , esp32-c2 , esp32-c3 , esp32-c5 , esp32-c6 , esp32-c61 , esp32-h2 , esp32-s2 , esp32-s3 , esp32-p4 , espressif systems , communication , ethernet , ip , wi-fi , thread , zigbee , matter , homekit , bluetooth , mqtt , adc , spi , uart , i2c , i2s , rmt , pwm , usb , usb otg , twai

• LamaPLC: Gas sensors	2023/07/01 15:29	gas, sensor, i2c, onewire, communication, mq-3, mq-4, mq-5, mq-6, mq-7, mq-8, mq-9, mq-135, gm-102b, gm-302b, gm-502b, gm-702b, alcohol, ch4, natural gas, smoke, lng, co, co2, lpg, h2, iso-butane, nox, nh3, benzene, town gas, formaldehyde, propane, humidity, temperature, voc, grv gas sens v2
• LamaPLC: GY-9250 MPU-9250/6500 9-axis Attitude Sensor Board	2026/02/14 22:45	ak8963, gy-9250, mpu-9250, 9-axis, motion detection, magnetometer, communication, i c, i2c, spi
• LamaPLC: HDC Texas Instruments Temperature/humidity sensors with I²C communication	2026/02/14 22:09	sht21, htu21, si7021, gy-21, gy-213v, hdc1080, gy-213v-hdc1080, cjmcu, cjmcu-1080, texas instruments, temperature, humidity, sensor, i2c, communication, arduino, code
• lamaPLC: HT16K33 display controller	2026/02/14 17:26	i2c, 7-segment display, display, ht16k33, arduino
• LamaPLC: HTU TE Connectivity temperature/humidity sensors with I²C communication	2026/02/14 21:54	htu, htu31d, htu21d, htu20d, sht20, htu20, sht21, htu21, si7021, gy-21, gy-213v, hdc1080, si702, gy-20, sht31, htu31, si7031, gy-31, te connectivity, temperature, humidity, i2c, communication, sensor, arduino, code
• lamaPLC: INA226 - current/voltage/power monitor with I²C communication	2026/02/14 23:58	i2c, i c, communication, arduino, energy, power, current, monitor, sensor, ina226, ina219, ina
• lamaPLC: LCD 1602/2004 with I²C communication	2026/02/14 17:27	communication, i2c, display, lcd, 1602, 2004, hd44780, pcf8574, pcf8574t, pcf8574at, arduino
• LamaPLC: MAX30100/MAX30102 Heart Rate Click Sensor Module	2026/02/14 23:38	max30102, max30100, heart rate click, sensor, communication, i2c, arduino, code
• lamaPLC: MCP23017 / MCP23S17 16-Bit I/O Expander with Serial Interface I²C / SPI	2026/02/14 22:52	communication, i2c, mcp23017, mcp23s17, spi, i o expander, serial, cjmcu-2317, cjmcu
• LamaPLC: Pixart PAJ7620U2 Gesture recognition sensors/module with I²C communication	2026/02/14 22:23	paj7620u2, gy-paj7620, pixart, gesture recognition, i2c, communication, sensor, arduino, code
• LamaPLC: SC16IS750 / SC16IS752: One or two serial (UART) ports from microcontroller via I²C or SPI communication	2026/02/14 22:53	cjmcu-750, cjmcu-752, cjmcu, nxp, sc16is750, sc16is752, uart, serial, i2c, spi, modul, converter, arduino, code
• LamaPLC: SGP Sensirion Gas-sensors with I²C communication	2026/02/15 20:27	sgp30, sgp40, sgp41, sensirion, gas-sensor, i2c, communication, sensor, arduino, code, eco2, voc, tvoc, indoor air quality, iaq, nox, hydrogen
• LamaPLC: SHT Sensirion Temperature/humidity sensor with I²C communication	2026/02/15 20:29	sht20, sht21, sht25, sht30, sht31, sht35, sht40, gy21, temperature, humidity, i2c, communication, sensor, arduino, code

- [lamaPLC: Signal level converters](#) 2026/02/14 22:47 [pca9306, i2c, voltage, level, converter](#)
- [lamaPLC: TCA9548A \(HW617\); Low-Voltage 8-Channel I²C Switch Module](#) 2026/02/14 22:51 [tca9548a, hw617, i2c, switch, communication, expansion board, arduino](#)
- [lamaPLC: TM1637 7-segment display](#) 2026/02/14 17:26 [i2c, 7-segment display, display, tm1637, arduino](#)
- [LamaPLC: TOFnnnC STMicroelectronics Time-of-Flight \(ToF\) sensors with I²C communication](#) 2026/02/14 22:22 [tof050c, vl6180, tof200c, vl53l0x, tof400c, vl53l1x, stmicroelectronics, time-of-flight, tof, i2c, communication, sensor, arduino, code](#)
- [LamaPLC: VL53Lnn STMicroelectronics time-of-flight \(ToF\) laser-ranging sensors with I²C communication](#) 2026/02/14 22:21 [vl53l0x, vl53l1x, vl53l0 1xv2, gy-530, time-of-flight, tof, laser-ranging, i2c, communication, sensor, arduino, code](#)
- [LamaPLC: VL6180X STMicroelectronics Time-of-Flight \(ToF\) sensor with I²C communication](#) 2026/02/14 22:22 [vl6180x, stmicroelectronics, time-of-flight, tof, i2c, communication, sensor, arduino, code](#)
- [Magnetic angle sensors](#) 2026/03/05 20:19 [magnetic angle sensor, magnetic flux, sensor, spi, i2c, pwm, communication, modul, as5047p, as5600, mt6701, mt6816, mt6835, tle5012b, amr, gmr, tmr, anisotropic magnetoresistive](#)
- [SSH1106/SSD1306 OLED Display with I²C communication](#) 2026/02/14 17:27 [i2c, oled, display, ssd1306, sh1106, ssh1106, arduino, cmos](#)

[I2c, I²C, SMBus, Philips, bus, communication, Arduino](#)

This page has been accessed for: Today: 2, Until now: 25

From: <https://lamaplc.com/> - **lamaPLC**

Permanent link: https://lamaplc.com/doku.php?id=com:basic_i2c

Last update: **2025/09/23 19:25**

