lamaPLC: TM1650 7-Segment Display with I²C like or Modbus Communication

The TM1650 is a popular, cost-effective integrated circuit (IC) used to drive 4-digit, 7-segment LED displays while minimizing the number of microcontroller I/O pins required.

The TM1650 7-segment display uses a proprietary 2-wire serial protocol that is very similar to I²C, but it is not a standard, addressable I²C device. It uses a clock (SCL) and a data line (SDA) to communicate with a microcontroller.

TM1650 Key Features

- **LED Display Controller:** Manages all multiplexing for a 4-digit display with optional decimal points.
- **Key Scanning:** It can also scan up to 4 keys/buttons.
- **Simple Interface:** Uses a 2-wire serial communication protocol that mimics the physical layer of I²C (SCL and SDA lines).
- **Adjustable Brightness:** Supports 8 intensity levels.
- **Low Pin Count:** Typically requires only VCC, GND, SCL, and SDA pins to operate.
- **Modbus RTU:** With TP8485E, it is an RS-485 communication transceiver

TM1650 Technical Specifications

Feature	Detail
Operating Voltage	3V to 5.5V
Display Digits	4 Digits, 8 Segments (including the decimal point)
Interface Type	2-wire serial (I²C-like) or with TP8485E RS-485 Modbus
Key Scan Capabilit	4-key input matrix
Brightness Level	8 levels

TM1650 and Arduino Wiring Diagram for I²C communication

- VCC to 5V
- GND to GND
- SCL (Clock) to A5 (or the SCL pin)
- SDA (Data) to A4 (or the SDA pin)

2026/02/14 23:38

TM1650 and Arduino I²C example code

```
#include <Wire.h>
#include <TM1650.h>

TM1650 d;

void setup() {
  Wire.begin();        // Start I²C communication
  d.init();            // Initialize the display
  d.displayOn();       // Turn display on
  d.setBrightness(4);  // Set brightness (0 to 7)
}

void loop() {
  // Display a fixed number
  d.displayString("1234");
  delay(2000);

  // Simple counter
  for (int i = 0; i <= 100; i++) {
    d.displayString(String(i).c_str());
    delay(100);
  }
}
```

Key Features

- **Brightness Control:** Use *setBrightness(level)* where level is 0–7.
- **Display Modes:** The TM1650 can operate in 8×4 or 7×4 display modes and supports keypress scanning, though some libraries focus primarily on the display output.

- **Individual Digits:** Use *setDot(digit, true)* to enable decimal points for specific digits.

TP8485E IC: RS-485 Transceiver

The TP8485E is a robust RS-485 transceiver designed for industrial communication networks.

- **Function:** It converts standard Transistor-Transistor Logic (TTL) or CMOS logic signals from a microcontroller into differential signals (A and B lines) suitable for long-distance, noise-resistant serial communication over twisted-pair cables.
- **Interface:** It communicates with a microcontroller via standard UART (Universal Asynchronous Receiver/Transmitter) pins (TX/RX) and requires additional General Purpose Input/Output (GPIO) pins to control the driver/receiver enable (DE/RE) lines.
- **Applications:** Its primary uses are in industrial automation, HVAC systems, e-metering networks, and DMX512 lighting control, where reliable data transmission over long distances in noisy environments is essential.

TM1650 and TP8485E Modbus Features

Modbus connection characteristics:
Default Modbus settings: 9600 baud, 8N1, RTU communication, slave ID: 1

After powering up, a number, such as **“0013”**, flashes briefly. The first digit represents the slave ID (**1**), and the second indicates the baud rate (**3: 9600**).

The “write multiple registers” Modbus function (10) works, with three pieces of information (signal, number of digits, brightness) (2.468 / low brightness 1: *TX 01 10 00 00 00 03 06 09 a4 00 03 00 01 a6 00*)

The registers must be written one at a time using the “write single register” function (6). (Send 1234: *TX: 01 06 00 00 04 d2 0b 57*)

Address	Function
0	To display Data send
1	Position of the decimal Point
2	Set Brightness
3	Set flash
4	Set Baudrate (Range 0 – 7 corresponds to 0:1200 1:2400 2:4800 3:9600 4:19200 5:38400 6:57600 7:115200)
5	Set the Address

TM1650 and TP8485E Modbus Wiring

TP8485E (RS-485):

- RO (Receiver Out) to Arduino RX (Pin 0 or SoftwareSerial RX).
- DI (Driver In) to Arduino TX (Pin 1 or SoftwareSerial TX).
- RE & DE (Enable pins) tied together to a digital pin (e.g., Pin 2) to toggle between Receive (LOW) and Transmit (HIGH).

TM1650 (Display):

- SCL to A5 and SDA to A4 (standard I2C pins).

TM1650 and TP8485E Modbus Arduino Example Code

This sketch receives a 4-digit number over RS-485 and displays it on a 7-segment display. You will need the TM1650 library by Anatoli Arkhipenko.

```
#include <Wire.h>
#include <TM1650.h>

#define RS485_CONTROL_PIN 2  // Pin to toggle RE/DE on TP8485E
TM1650 d;

void setup() {
  // Initialize RS-485
  Serial.begin(9600);
  pinMode(RS485_CONTROL_PIN, OUTPUT);
  digitalWrite(RS485_CONTROL_PIN, LOW); // Set to RECEIVE mode initially

  // Initialize TM1650 Display
  Wire.begin();
  d.init();
  d.displayOn();
  d.setBrightness(4);
  d.displayString("WAIT");
}

void loop() {
  if (Serial.available() > 0) {
    // Read incoming string from RS-485 (e.g., "1234")
    String receivedData = Serial.readStringUntil('\n');
    receivedData.trim(); // Clean whitespace

    if (receivedData.length() > 0) {
      d.displayString(receivedData.c_str());
    }
  }
}

// Function to send data back via RS-485 if needed
void sendRS485(String msg) {
  digitalWrite(RS485_CONTROL_PIN, HIGH); // Switch to TRANSMIT mode
  Serial.println(msg);
  Serial.flush();                        // Ensure all data is sent
  digitalWrite(RS485_CONTROL_PIN, LOW);  // Switch back to RECEIVE mode
}
```

- **Mode Toggling:** For half-duplex chips like the TP8485E, you must set the DE/RE pins HIGH to send data and LOW to listen.
- **Power Stability:** TM1650 displays can be “noisy” on the 5V rail; adding a 100µF capacitor across VCC and GND near the display is highly recommended to prevent flickering.
- **Addressing:** If your RS-485 bus has multiple devices, you should implement a simple protocol (e.g., *ID:DATA*) so the Arduino displays only messages intended for it.

Display topics on lamaPLC

Page	Date	Tags
• lamaPLC: DM56A04 / DM36B06 digital tube display with Modbus Communication	2026/02/14 18:25	dm56a04, dm36b06, eletechsup, 7-segment, display, modbus, rtu, modbus rtu, arduino
• lamaPLC: HT16K33 display controller	2026/04/23 21:51	i2c, 7-segment display, display, ht16k33, arduino
• lamaPLC: LCD 1602/2004 with I²C communication	2026/02/14 18:27	communication, i2c, display, lcd, 1602, 2004, hd44780, pcf8574, pcf8574t, pcf8574at, arduino
• lamaPLC: st756x display drivers	2026/05/20 16:17	display, driver, i2c, spi, lcd, cog, oled, st7565, st7567, gm12864, gm12864-59n, gm12864-03a, gm12864-01a, gme12864-41
• lamaPLC: TM1637 7-segment display	2026/02/14 18:26	i2c, 7-segment display, display, tm1637, arduino
• lamaPLC: TM1650 7-Segment Display with I²C like or Modbus Communication	2026/02/14 18:26	tm1650, stc8g, tp8485e, hyuduo5x1b64edtk1244, 7-segment, display, modbus, rtu, modbus rtu, arduino
• SSH1106/SSD1306 OLED Display with I²C communication	2026/02/14 18:27	i2c, oled, display, ssd1306, sh1106, ssh1106, arduino, cmos

TM1650, STC8G, TP8485E, Hyuduo5x1b64edtk1244, 7-segment, display, Modbus, RTU, Modbus RTU, Arduino