CS LT communication using UART protocol on Arduino board

1. Introduction

This document describes the implementation of a UART (Universal Asynchronous Receiver/Transmitter) communication protocol to interface an Optris CS LT series non-contact infrared temperature sensor with an Arduino Uno R4 board. The system reads temperature data and sensor parameters, displaying them on a 16x2 character LCD.

The primary objectives of this integration are to:

- Read the process temperature (Tproc) and ambient temperature (Tamb).
- Read and write sensor configuration parameters (emissivity and transmissivity).
- Display the measured values and parameters in real-time on the LCD.

2. Setup

2.1. Components

- Controller: Arduino UNO R4
- Temperature Sensor: Optris CS LT Series
- Display: Allnet 4duino Display Modul LCD 1602
- Passive Component: 100kΩ Resistor

2.2. Preparation for CS LT

Before physical integration with the Arduino, the sensor must be configured for UART communication.

1. **Hardware Connection:**
   - Connect the CS LT sensor to a computer using the Optris IR App Connector.
2. **Software Configuration:**
   - Launch the **Optris CompactConnect** software on the computer.
   - Navigate to **SETUP**.
   - Set the **Mode** under the Tab **IN/OUT** to "Communication Input".
   - Set the **Mode** under the Tab **OUT** to "Communication Output".
3. **Finalization:** Close the software and disconnect the sensor from the computer.

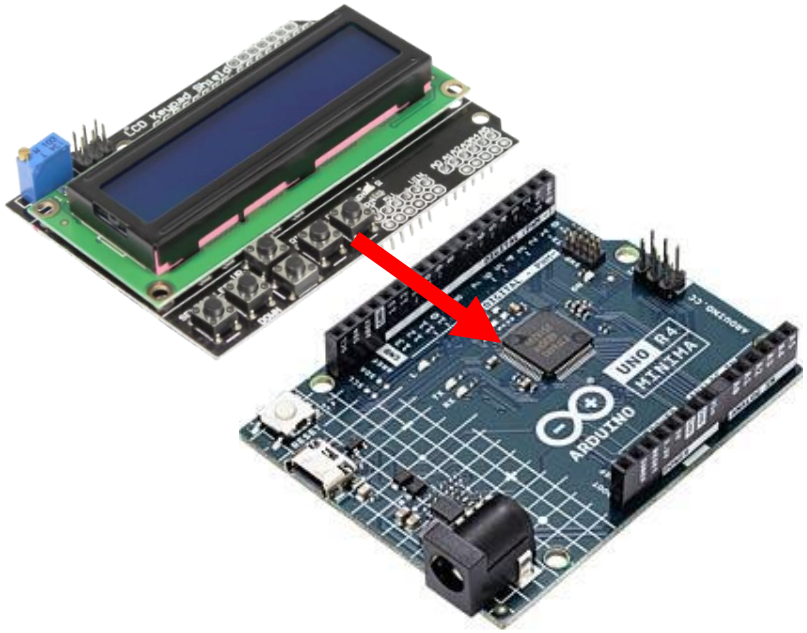
Important: This configuration step is mandatory to enable the sensor's serial command interface. Only after successful configuration should the sensor be connected to the Arduino board.

Optris GmbH & Co. KG
Ferdinand-Buisson-Str. 14
13127 Berlin
Tel. +49 (0)30 500 197-0
Fax +49 (0)30 500 197-10
info@optris.com
www.optris.com

2.3. Connections and Mechanical Installation

The Optris CS LT sensor operates at 3.3 V logic levels. The needed power supply for the sensor is at least 5V. Use the USB cable to connect the Arduino board to your computer to compile the source code and upload the code to the board. For operating the sensor and the display use the power supply of 12 V.

The LCD must be connected to the specified digital pins as per the LiquidCrystal object initialization in the code (there is only one way possible).



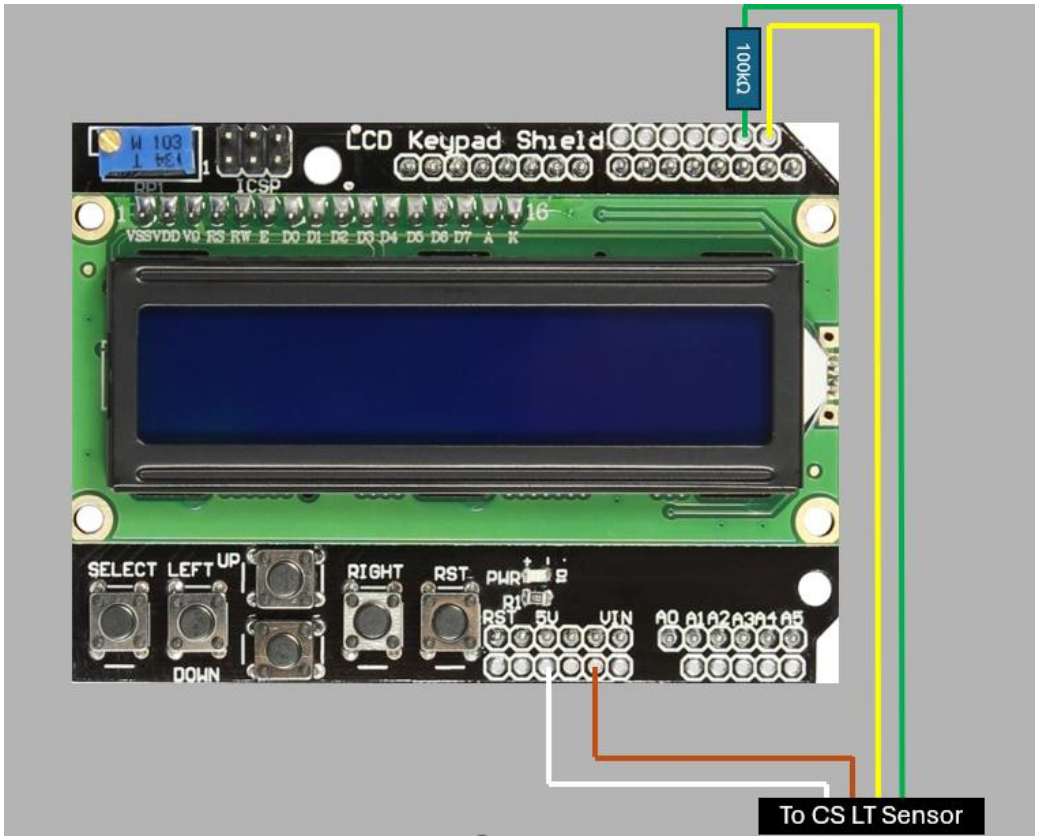
Picture 1: LCD Mounting

Connect the CS LT Sensor to the LCD using the pin mapping provided in the table or in the picture below.

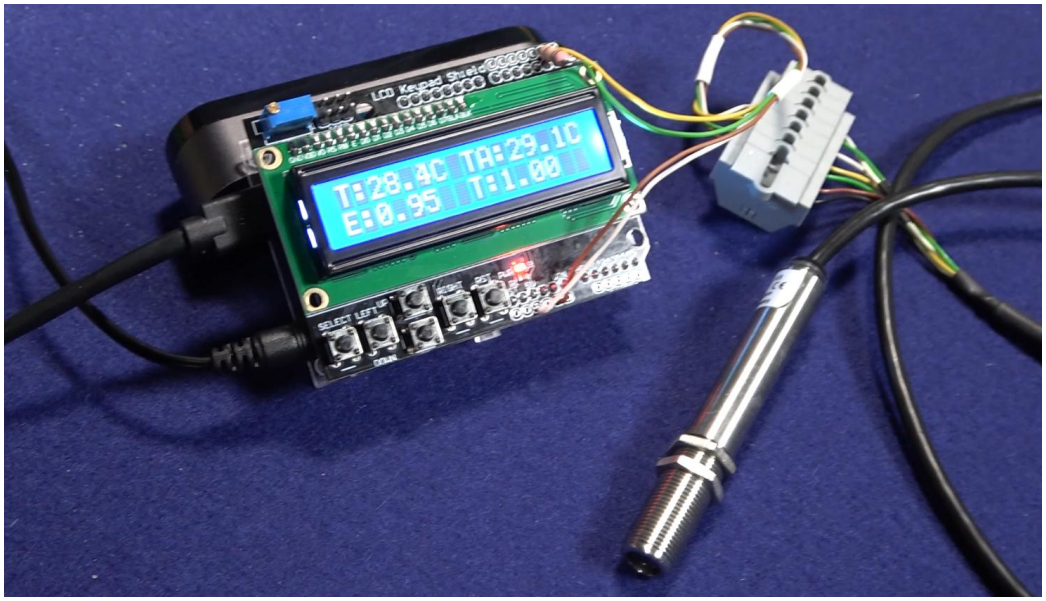
Arduino	Display	Optris CS LT Sensor
5V	5V	White (Power)
GND	GND	Brown (GND)
TX→1	2nd upper right PIN	IN/OUT -> Green (RxD)[1)]
RX←0	1st upper right PIN	OUT -> Yellow (TxD)

[1)] Note: The connection from Arduino Pin 1 (TX) to the sensor's Green (RxD) wire requires a 100 kΩ resistor to be installed in parallel. Please ensure this is correctly implemented to adhere to the sensor's logic level specifications.

Optris GmbH & Co. KG
Ferdinand-Buisson-Str. 14
13127 Berlin
Tel. +49 (0)30 500 197-0
Fax +49 (0)30 500 197-10
info@optris.com
www.optris.com



Picture 2: Wiring Diagram



Picture 3: Connected CS LT to the Arduino and LCD

Optris GmbH & Co. KG
Ferdinand-Buisson-Str. 14
13127 Berlin
Tel. +49 (0)30 500 197-0
Fax +49 (0)30 500 197-10
info@optris.com
www.optris.com

3. Compiling and uploading

- Open the provided source code in the Arduino IDE (or your preferred Arduino-compatible software).
- Connect the Arduino Uno R4 to your computer via the USB cable.
- In the IDE, select the correct Board (Arduino Uno R4) and the corresponding COM Port.
- Compile the source code and upload it to the Arduino board.

4. System Operation

For running the sensor and display, provide a 12V power supply to the Arduino board's power jack. Upon successful upload, the system will automatically begin operation. Now you have successfully established the UART communication between the Optris CS LT and the Arduino Board. The following functionalities are now active:

LCD Output: The main loop continuously reads and displays Tproc, Tamb, Emissivity, and Transmissivity on the LCD.
Button Control: The system polls an analog input connected to a resistor ladder for button presses.
UP/DOWN Buttons: Increment/Decrement emissivity by 0.01 (range: 0.11 to 1.10).
LEFT/RIGHT Buttons: Decrement/Increment transmissivity by 0.01 (range: 0.11 to 1.10).
SELECT Button: Displays the sensor's serial number on the LCD.

4.1. Additional Sensor Functions

For advanced functionality and a complete list of available commands, please consult the accompanying command list ***CS/CSmicro Communication interface***. Or download the list from *https://optris.com/wp-content/uploads/2025/05/CS-CSmicro-commands-2025-05.pdf*

5. Source Code

```
#include <LiquidCrystal.h>                // include the LCD library code

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);      // Creates an LCD object. Parameters: (rs,
enable, d4, d5, d6, d7)

#define BUTTON_RIGHT  0
#define BUTTON_UP     1
#define BUTTON_DOWN   2
#define BUTTON_LEFT   3
#define BUTTON_SELECT 4
#define BUTTON_NO     5

byte Bytes[16];                                  // byte array for send and receive
float Tproc, Tamb, Emiss = 0, Trans = 0;  // calculated temperature, ambient
temperature, emissivity, transmissivity
unsigned long SerialNumber = 0;

void setup() {
  Serial1.begin(9600);              // initialize Optris CS UART with baud rate of
9600 bps
  Serial1.setTimeout(50);           // set read timeout to 50ms
  lcd.begin(16, 2);                 // set up the LCD's number of columns and rows:
  lcd.clear();                      // Clears the LCD screen
```


Optris GmbH & Co. KG
Ferdinand-Buisson-Str. 14
13127 Berlin
Tel. +49 (0)30 500 197-0
Fax +49 (0)30 500 197-10
info@optris.com
www.optris.com

```
  SerialNumber = readSerial();      // lcd.setCursor(0, 0);
// set the cursor to column 0, line 1 (note: line 1 is the second row, since
counting begins with 0):)
  // lcd.print("Opt-CS #"); lcd.print(SerialNumber);
// Print a message to the first line from LCD.
  writeEmiss(0.95);                 // set Optris CS emissivity to 0.95
  writeTrans(1.000);               // set Optris CS transmissivity to 1.000
  Emiss = readEmiss();             // read "Emissivity" from Optris CS
  Trans = readTrans();             // read "Transmissivity" from Optris CS
}

int readSwitches() {                      // check if a button is pressed
  int analogValue = analogRead(A0);    // read the analog value for the buttons.
  if (analogValue > 1000) return BUTTON_NO;
  if (analogValue <   50) return BUTTON_RIGHT;
  if (analogValue <  195) return BUTTON_UP;
  if (analogValue <  380) return BUTTON_DOWN;
  if (analogValue <  555) return BUTTON_LEFT;
  if (analogValue <  790) return BUTTON_SELECT;
  return BUTTON_NO;
}
// Optris CS routines
//--------------------
float readTproc() {                           // read the Tprocess value
  Serial1.write(0x01);                      // send 0x01 to Optris CS
  if (Serial1.readBytes(Bytes,2) == 2) {     // try to receive answer of 2 bytes
    return (float)(Bytes[0]*256 + Bytes[1]) / 10. -100.; // calulate temperature
  } else return -100;
}

float readTamb() {                            // read the Tambient value
  Serial1.write(0x02);                       // send 0x02 to Optris CS
  if (Serial1.readBytes(Bytes,2) == 2) {     // try to receive answer of 2 bytes
    return (float)(Bytes[0]*256 + Bytes[1]) / 10. -100.;   // calulate temperature
  } else return -100;
}

float readEmiss() {
  Serial1.write(0x04);                       // send 0x04 to Optris CS
  if (Serial1.readBytes(Bytes,2) == 2) {     // try to receive answer of 2 bytes
    return (float)(Bytes[0]*256 + Bytes[1]) / 1000.; // calculate emissivity
  } else return 0;
}

float readTrans() {
  Serial1.write(0x05);                          // send 0x05 to Optris CS
  if (Serial1.readBytes(Bytes,2) == 2) {     // try to receive answer of 2 bytes
    return (float)(Bytes[0]*256 + Bytes[1]) / 1000.; // calculate emissivity
  } else return 0;
}

void writeEmiss(float emiss) {
  int optemiss = emiss *1000;
  Bytes[0] = 0x84;
  Bytes[1] = optemiss / 256;
  Bytes[2] = optemiss % 256;
  Bytes[3] = Bytes[0] ^ Bytes[1] ^ Bytes[2];   // calculate checksum
  Serial1.write(Bytes,4);                    // send 0x84 to Optris CS
```


Optris GmbH & Co. KG
Ferdinand-Buisson-Str. 14
13127 Berlin
Tel. +49 (0)30 500 197-0
Fax +49 (0)30 500 197-10
info@optris.com
www.optris.com

```
  Serial1.readBytes(Bytes,2);
}
void writeTrans(float trans) {
  int opttrans = trans *1000;
  Bytes[0] = 0x85;
  Bytes[1] = opttrans / 256;
  Bytes[2] = opttrans % 256;
  Bytes[3] = Bytes[0] ^ Bytes[1] ^ Bytes[2];    // calculate checksum
  Serial1.write(Bytes,4);                      // send 0x85 to Optris CS
  Serial1.readBytes(Bytes,2);
}
unsigned long readSerial() {                     // read the serial number
  Serial1.write(0x0e);                         // send 0x0e to Optris CS
  if (Serial1.readBytes(Bytes,4) == 4) {  // try to receive answer of 4 bytes
    unsigned long serial = 0;
    serial += Bytes[0]; serial <<= 8;
    serial += Bytes[1]; serial <<= 8;
    serial += Bytes[2]; serial <<= 8;
    serial += Bytes[3];
    return serial;
  } else return -1;
}
// Loop
void loop() {
  Tproc = readTproc();                     // read "Tproccess" from Optris CS
  Tamb = readTamb();                    // read "Tambient" from Optris CS
  Emiss = readEmiss();
  Trans = readTrans();
  lcd.setCursor(0, 0);                     // set the cursor to column 0, line 1
(note: line 1 is the second row, since counting begins with 0):)
   lcd.print("T:"); lcd.print(Tproc,1); lcd.print("C TA:"); lcd.print(Tamb,1);
lcd.print("C");     // Print Tprocess and Tambient to LCD.
  lcd.setCursor(0, 1);              // set the cursor to column 0, line 1
  lcd.print("E:"); lcd.print(Emiss); lcd.print("  T:"); lcd.print(Trans);
lcd.print("  ");                          // Print a message to the 1st line LCD.
  switch(readSwitches()) {
    case BUTTON_UP :
      if (Emiss <= 1.09) { Emiss += 0.01; writeEmiss(Emiss); Emiss = readEmiss(); }
      break;
    case BUTTON_DOWN :
      if (Emiss >= 0.11) { Emiss -= 0.01; writeEmiss(Emiss); Emiss = readEmiss(); }
      break;
    case BUTTON_LEFT :
      if (Trans <= 1.09) { Trans -= 0.01; writeTrans(Trans); Trans = readTrans(); }
      break;
    case BUTTON_RIGHT :
      if (Trans >= 0.11) { Trans += 0.01; writeTrans(Trans); Trans = readTrans(); }
      break;
    case BUTTON_SELECT :
      lcd.setCursor(0, 0);
      lcd.print("OPTRIS CS-SENSOR");
      lcd.setCursor(0, 1);
      lcd.print("Serial #"); lcd.print(SerialNumber);
      break;
  }
  delay(150);                              // wait 50ms
}
```


Optris GmbH & Co. KG
Ferdinand-Buisson-Str. 14
13127 Berlin
Tel. +49 (0)30 500 197-0
Fax +49 (0)30 500 197-10
info@optris.com
www.optris.com