

# AN-014 UART Interface and Arduino Support

With the release of v1.3.0 firmware, the OPS241 and OPS242 short range radar sensors support data transfer over the UART interface. This enables the usage of the sensors in several new and different applications including directly talking to an Arduino board. This application note will describe the operation of the UART interface and show how to interface an OPS241 and OPS242 to an Arduino board.

## UART Interface

Starting with v1.3.0 firmware release, the OPS241 and OPS242 support data communication over UART. The sensor default operation powers up with UART enabled. If USB is enumerated, the UART will be stopped and all data will pass out the USB interface. The basic configuration of the UART is listed in Table 1 with the default UART baud rate set at 19,200 bps.

Table 1. UART Specifications

Parameter	Specification
Baud Rate	19,200
Bits	8
Parity	None
Stop Bit	1

The API and data formats are identical between the UART and USB interfaces. See Application Note [AN-10 API Interface](#) for API updates in v1.3.0. New to the API is the ability to support baud rates ranging from 9,600 to 230,400 bps. To change the baud rate, write the *In* command to the OPS241/OPS242 where *n* is 1 to 5 as shown in

Table 2. Writing *I?* to the OPS241/OPS242 will report the baud rate setting.

Table 2. UART Baud Rate API

Parameter	Specification
I1	9,600
I2	19,200 (default)
I3	57,600
I4	115,200
I5	230,400
I?	Query Baud Rate

When connecting the UART signals of the OPS241 and OPS242, make sure the voltage levels and the connections of the signals with the host system are correct. Figure 1 provides the basic connection

guidance. Note that the UART on recent OPS241 support 5V signaling (early versions were 3.3V only) while the OPS242 only supports 3.3V operation.

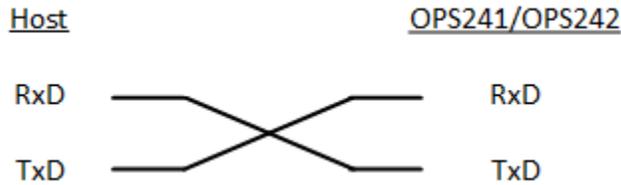


Figure 1. UART Pin Connections

The OPS241 and OPS242 can easily be connected to an embedded processor board or Raspberry Pi. For direct connections to Arduino boards, special considerations should be considered as described below.

### OPS241 Arduino UART Interface

The OPS241 was designed as a shield board with the capability to plug directly into an Arduino board. Depending on the board you have and how you want to use the UART, there are three different configurations to consider. Table 3 summarizes these different configurations. The below guidelines describe connecting the OPS241 to an Arduino Uno or Leonardo board. These instructions may need to be modified if connecting to one of the other Arduino boards such as Mega and Due.

Table 3. OPS241 Board UART Configuration

Option	Board Serial Number	Full UART Operation	5V Signals	UART Arduino Connection	Header Installation
A	< 1000	TxD Only	No	None	None (use TP 34)
B	≥ 1000	Yes	Yes	Software Serial	J1/J4
C	≥ 1000	Yes	Yes	Hardware Serial	J1/J5

Early OPS241 boards (< 1000 serial number) do not support 5V UART signal levels. You can read the serial number from the barcode on the back of the board. The last 4 digits on the right are the board serial number (Figure 2).

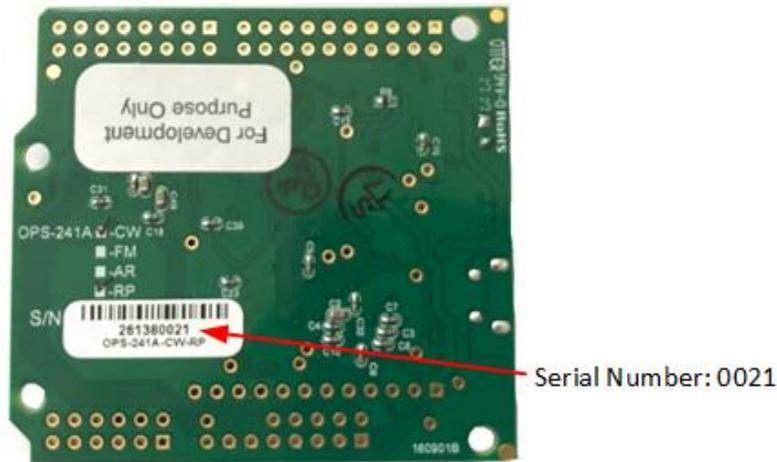


Figure 2. Board Serial Number

These early boards can only be used as UART Tx/D devices by connecting into the 3.3V signal at test point 34 (TP34). This signal is 3.3V and should not be directly connected to Arduino which uses 5V signals. A level shifter may be used between TP34 and the Arduino Rx/D pin to send the serial data. Alternatively, some Arduino are designed with 3.3V signal levels(see Appendix).

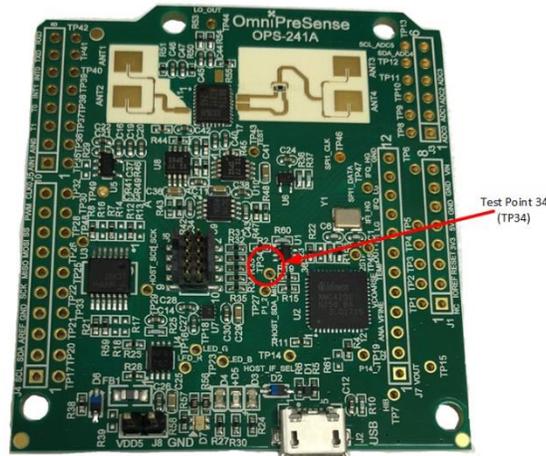


Figure 3. OPS241 TP34 Location

Starting with serial number 1000, the circuitry to enable plugging an OPS241 directly into an Arduino has been enabled. These OPS241 are shipped with two headers which can be soldered into place for direct connection into an Arduino board. The basic configuration utilizes SoftSerial to implement the UART on the Arduino. In this configuration, headers are soldered into connectors J1 (8-pin) and J4 (10-pin) as shown in Figure 4. The headers should be soldered into the underside of the board with the long end of the pins facing down. The J1 header provides 5V power and ground connections while J4 provides the UART interface on pins 6 (MISO/TxD) and 7 (MOSI/RxD). The pin 6 TxD communicates with pin 12 on the Arduino header and pin 7 RxD communicates with pin 11.



Figure 4. Basic Arduino Header Installation (Option B)

The SoftwareSerial library is utilized in this configuration. The below reference code enables SoftwareSerial with the default baud rate of 19,200. The RxD and TxD pins are assigned to match the TxD and RxD output from the OPS241. This code sends the data out over the USB interface. Do not echo back data on the TxD/RxD lines as it will corrupt the USB data.

```

/*
  Software serial example code.
  */
#include <SoftwareSerial.h>
SoftwareSerial ops241Serial(12, 11); // RX, TX

void setup() {
  Serial.begin(19200);

  // set the data rate for the SoftwareSerial port
  ops241Serial.begin(19200);
}

void loop() { // run over and over
  if (ops241Serial.available()) {
    Serial.write(ops241Serial.read());
  }
  if (Serial.available()) {
    ops241Serial.write(Serial.read());
  }
}

```

Option C in Table 3 makes use of the hardware UART on the Arduino. This allows the faster, native microprocessor code to handle the serial communication. To enable the hardware UART, two additional wires need to be soldered to the board and the header should be populated on J5 instead of J4. First, populate the header in J1 as before. This will provide power and ground to the OPS241. Next, take the 10-pin header provided and remove two of the pins to make it an 8-pin header. This can be done by easily

bending the plastic of the last two header pins or cutting them off. Solder in the two 8-pin headers into J1 and J5 to the bottom of the OPS241 with the long end of the header facing down as shown in Figure 5.



Figure 5. Arduino Hardware Serial (Option C)

Solder one jumper wire from TP25 (next to pin 6 of J4) to TP42 (next to pin 8 of J5). This will match up to RxD on the Arduino. Solder a second jumper wire from TP26 (next to pin 7 of J4) to TP41 (next to pin 7 of J5). This will match up to TxD on the Arduino. Once these are done, the OPS241 can be directly plugged into the Arduino.

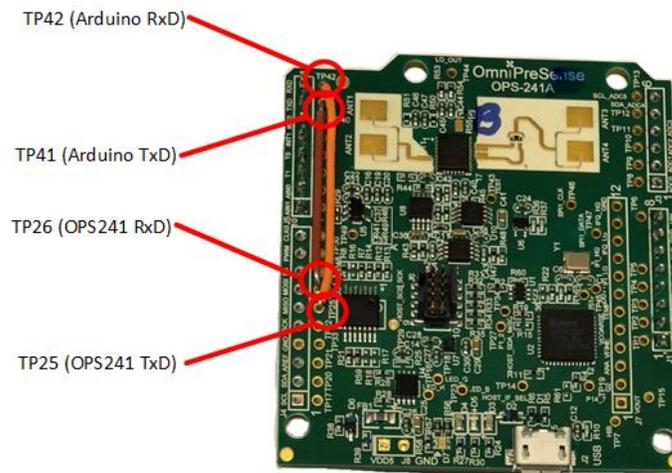


Figure 6. Arduino Hardware UART Jumper Wire Soldering

Basic reference code for enabling the hardware UART is available below.

```
void setup() {  
  //Initialize serial and wait for port to open:
```

```

Serial.begin(19200);
}
void loop() {
    float v = Serial.parseFloat();
    // v is the speed as reported by the radar sensor
}

```

## OPS242 UART Interface

On the OPS242, the UART interface is provided on the J8 connector. These pins are 3.3V signal levels only. They should not be connected directly to an Arduino with 5V signal levels unless a level shift is provided. The pins assignments are as listed in

Table 4.

Table 4. OPS242 UART Pin Assignment

Pin	Direction	Pin Name	Pin Description
1	-	NC	No Connect
2	-	NC	No Connect
3	Output	Host Interrupt	Interrupt from OPS242 to system
4	Input	/Reset	System reset of OPS242 (active low)
5	Input	SPI SEL	SPI device select
6	Input	SPI MOSI/RxD	SPI data input/UART receive
7	Output	SPI MISO/SDA/TxD	SPI data output/I <sup>2</sup> C data/UART transmit
8	Input	SCL/SCK	SPI/I2C clock
9	-	5V	Power
10	-	Ground	Ground

A simple check of the UART interface is to use a UART to USB converter board such as an FTDI board from [SparkFun](#). Connect the TxD and RxD pins of the OPS242 to the converter board and the USB cable to a PC. Open a simple Command Terminal such as [Tera Terminal](#) and [PuTTY](#). The data from the OPS242 will automatically show up in the terminal window and API commands can be sent back to query or configure the board.

## Appendix

### Arduino Serial Port Configurations

Arduino Board	TxD/RxD	Pins	Voltage	Notes
Uno	RxD	0	5V	
	TxD	1	5V	
Leonardo	RxD	0	5V	
	TxD	1	5V	
Mega	RxD	0	5V	
	TxD	1	5V	
	RxD	19	5V	Serial1
	TxD	18	5V	Serial1
	RxD	17	5V	Serial2
	TxD	16	5V	Serial2
	RxD	15	5V	Serial3
	TxD	14	5V	Serial3
Due	RxD	0	3.3V	
	TxD	1	3.3V	
	RxD	19	3.3V	Serial1
	TxD	18	3.3V	Serial1
	RxD	17	3.3V	Serial2
	TxD	16	3.3V	Serial2
	RxD	15	3.3V	Serial3
	TxD	14	3.3V	Serial3

## Revision History

Version	Date	Description
A	October 23, 2018	Initial release.