AN-014 UART Interface and Arduino Support

All OPS24x sensors support UART interface which provides a simple connection to Arduino. This application note will describe the operation of the OPS24x UART interface and show how to interface it to Arduino boards.

UART Interface

All OPS24x sensors default operation power up with UART enabled. If USB is enumerated, the UART will be stopped and all data will pass out the USB interface. UART interface can be restarted with the API command IR which can be set to persistent memory with the A! command. 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 sensor API and data formats are identical between the UART and USB interfaces. See Application Note AN-10 API Interface for details on the API control of the output and radar signal processing. The API has the ability to support baud rates ranging from 9,600 to 230,400 bps. To change the baud rate, write the I*n* command to the OPS24x where *n* is 1 to 5 as shown in Table 2. Writing I? to the OPS24x 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 OPS24x, make sure to connect the sensor TxD to the host RxD and the sensor RxD to the host TxD (Figure *1*). The UART voltage levels are different on the OPS24x and Arduino and care must be taken to make sure the right voltage levels are being used.

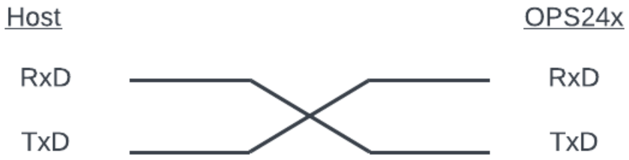

Figure 1. UART Pin Connections

The OPS241 supports 5V UART signals and direct connection to original Arduino such as the Uno, Leonardo, or Mega. The OPS242 and OPS243 have 3.3V UART signals and can make a direct connection to an Arduino Due with its 3.3V UART but require a level shifter to support any of the 5V UART Arduino boards. See the Appendix for the list of Arduino boards and their UART voltage levels.

OPS241 Arduino Interface

The OPS241 was designed as an Arduino shield board and can be directly plugged into an Arduino. The software serial port on the UART interface is used by the OPS241. A 10-pin header can be soldered to J4 along with and 8-pin header to J1 to provide a direct plug-in connection with an Arduino (Figure 2). Header J1 provides 5V power and ground to the OPS241.

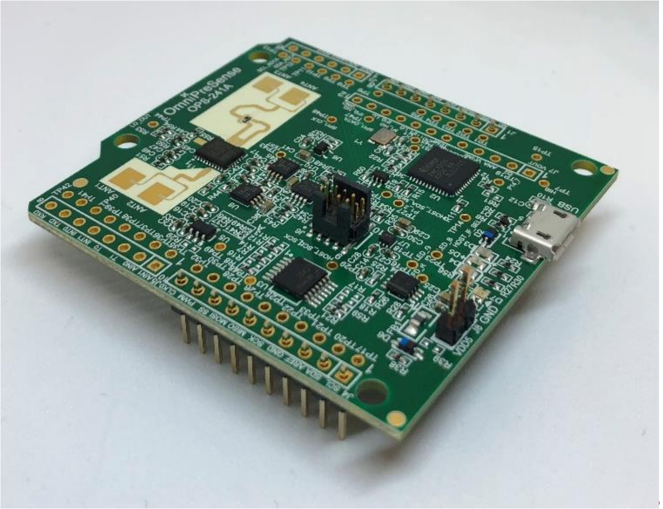

Figure 2. OPS241 with Header J1 and J4 Installed

The SoftwareSerial library is utilized for serial communication. The reference code below enables SoftwareSerial with the default baud rate of 19,200. The RxD and TxD pins are assigned to match the TxD and RxD of the OPS241. This code also 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. Conflicts have been seen with UART operation when USB is used for power. It's recommended to use the barrel connector of the Arduino for power for proper UART operation.

```
/*
  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());
  }
}
```

If desired, it's possible to make use of the Arduino hardware serial pins by wiring them to the same interface as the software serial. Hardware serial 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 J5 header should be populated.

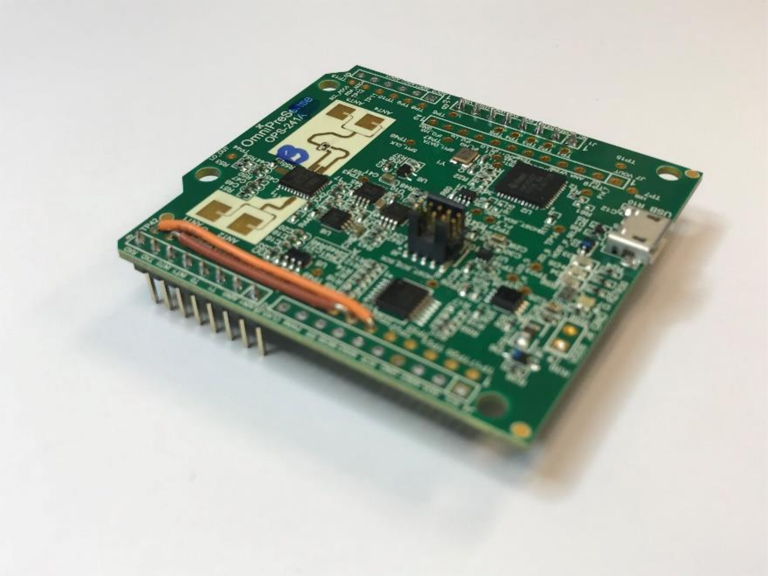

Figure 3. Arduino Hardware Serial with Header J5 Installed

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.

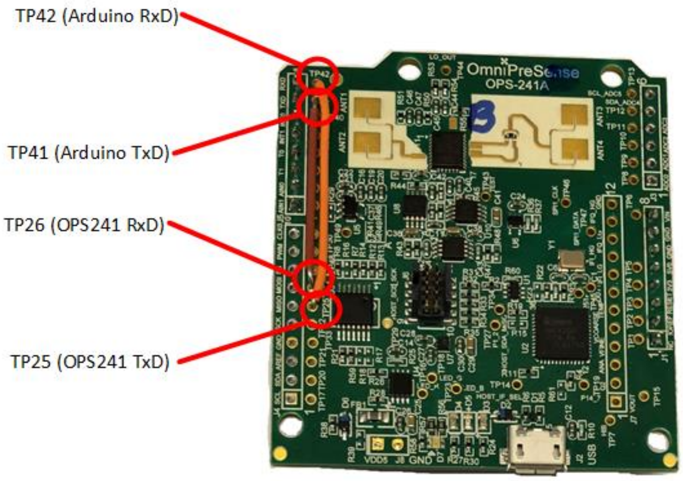

Figure 4. Arduino Hardware UART Jumper Wire Soldering

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 OPS241-A radar sensor or range by OPS241-B
}
```

OPS242 Arduino Interface

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

Table 3. For UART support, connect the TxD and RxD on the OPS242 with the host system keeping in mind the swapping of names between boards as shown in Figure 1. In addition, make sure to connect the 5V and Ground pins on J8. Maximum power draw on the OPS242 5V will be 1.52W or approximately 302mA. It may require its own power source.

Table 3. 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 (not used)
6	Input	SPI MOSI/RxD	SPI data input/UART receive
7	Output	SPI MISO/SDA/TxD	SPI data output/I^{2}C data/UART transmit
8	Input	SCL/SCK	SPI/I2C clock (not used)
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.

OPS243 Arduino Interface

The OPS243, like the OPS242, has 3.3V UART interface and should only be used with the same voltage level boards (ex. Due) or require a level shifter for 5V UART boards (ex. Uno). The OPS243 UART interface is on the 10-pin J3 header. The OPS243 (non-WiFi versions) provides an option to use a jumper setting to enable RS-232 instead of UART interface. UART is the default setting but make sure that jumper J8 is set correctly for UART operation. See the OPS243 detailed datasheet for further details.

An example of the OPS243 wired to an Arduino Due is provided in Figure 5 showing the direct connection. The OPS243 draws too much current and should be provided with its own power source.

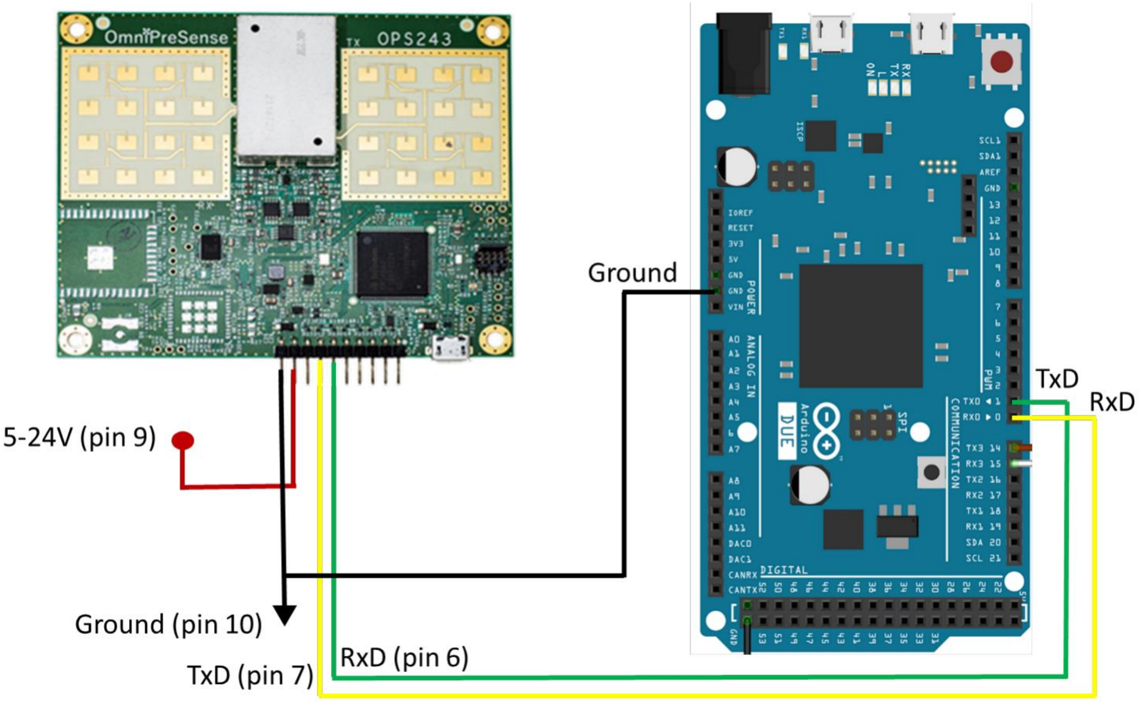

Figure 5. OPS243 Connection to Arduino Due

If it's desired to connect the OPS243 to an Arduino with 5V UART interface, a level shifter may be implemented between the boards as shown in Figure 6. A low cost level shifter such as the Sparkfun BOB-12009 can be utilized with 5V power provided by the Arduino and separate 3.3V power source required.

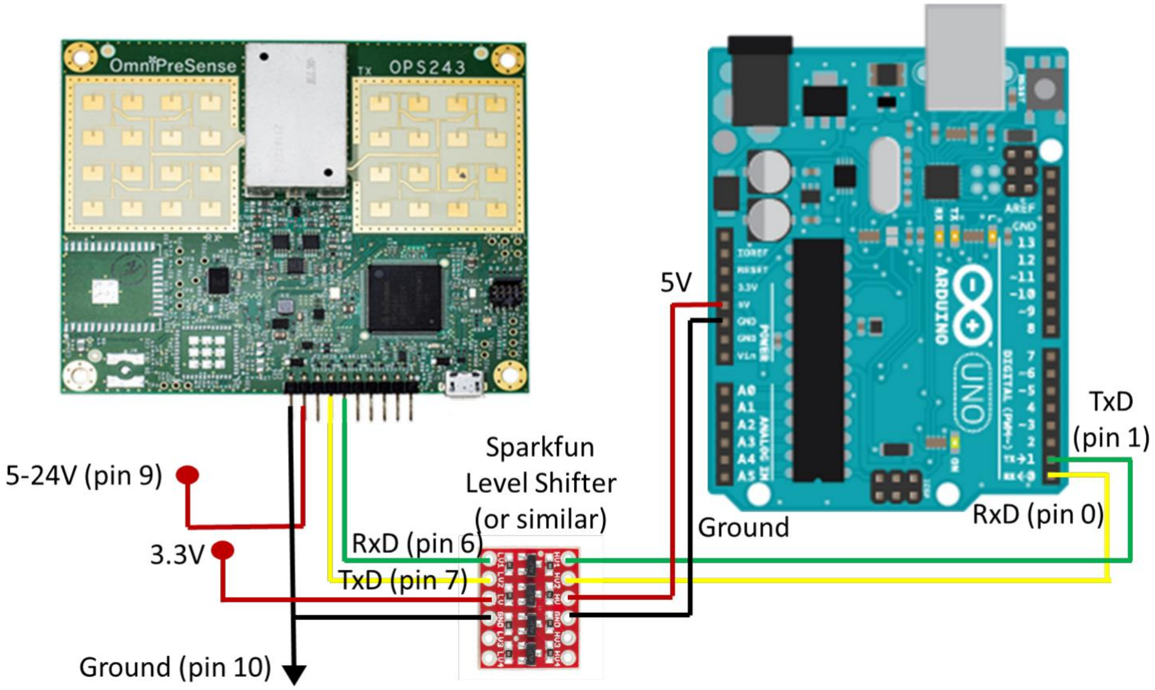

Figure 6. OPS243 Connection to Arduino Uno with Level Shifter

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.
B	November 20, 2018	Added instructions for enabling 5V UART on older OPS241.
C	June 27, 2024	Updated information including OPS243 support. Removed reference to very old OPS241 board support.