AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

ABSTRACT

The purpose of this application note is to explain how to connect a GSS CO_2 sensor to an Arduino board. Arduino is an open-source hardware and software company that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. There are number of different hardware options available, and most allow easy connection to a GSS CO_2 sensor.

GSS sensors have a variety of methods to communicate with an external microcontroller, including UART and I^2C digital serial ports, as well as legacy analogue interfaces.

For the purposes of this application note, it will concentrate on the popular UART interface although similar principles apply when using the I^2C interface.

The sample code implements a software UART, using two general purpose digital I/O pins on the Arduino UNO to communicate with the GSS sensor. All data from this UART is passed to the Arduino UNO hardware UART and allows data to be transferred to PC via the Arduino USB connection. The connection diagram works with the sample code, but it is possible to use other pins, if the corresponding modifications are made to the code.

AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

TABLE OF CONTENTS

AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

BASIC PRINCIPLES

A UART interface, or universal asynchronous receiver-transmitter, is one of the most common device-to-device communication protocols. The UART interface consists of two signals, the Transmitter (Tx) and Receiver (Rx), which transmit and receive serial data intended for serial communication. All current GSS gas sensors have an UART interface.

With the exception of the CozIR-Blink, which must be power-cycled to enable normal operation, all GSS gas sensors include a dedicated UART interface that can be used to easily connect to an Arduino board without any additional hardware.

CONNECTING TO AN ARDUINO

The Arduino has a number of re-configurable digital I/O pins. In this case, the example makes use of the SoftwareSerial driver built into the Arduino software to enable two digital pins to emulate a UART interface. The Arduino board must be connected to the GSS sensor UART interface. The Arduino board is also capable of supplying ~400mA of power on the 3.3V supply pin, sufficient for any GSS sensor.

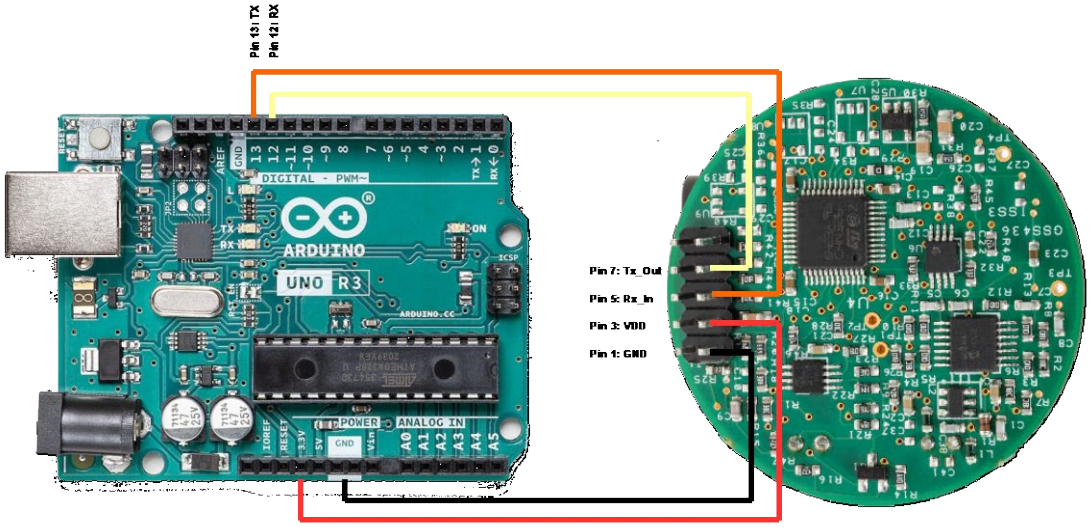
The table below shows the connections for the Arduino Uno board and a CozIR-A CO_2 sensor.

Arduino Uno			**GSS CozIR-A**		
Pin	**Name**	**Description**	**Pin**	**Name**	**Description**
13	Tx	Digital Output	5	Rx_In	UART Receive
12	Rx	Digital Input	7	Tx_Out	UART Transmit
4	+3V3	Supply	3	VDD	Sensor Supply
6 or 7	GND	Ground	1	GND	Sensor Ground

As soon as power is applied to a GSS gas sensor, and the UART connection is enabled, it automatically begins to report gas concentration levels without any user intervention. The exception to this rule is the CozIR®-Blink, which will automatically report a single measurement only. The CozIR®-Blink will cease to report further measurements until it has been power-cycled.

AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

CONNECTING THE ARDUINO UNO TO THE COZIR-A SENSOR



SOFTWARE STRUCTURE

The example software acts as a bridge between the sensor UART and the Arduino UNO UART / USB connection to the HOST PC. The software consists of a free running loop, continuously checking for characters being received from either of the two UART interfaces on the Arduino UNO. All data received is transmitted to the alternate UART without any interpretation or modification. Two buffers are used to store ASCII characters temporarily, allowing the UARTs to operate at different baud rates.

AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

SAMPLE CODE FOR COZIR-A AMBIENT SENSOR

```
#include <SoftwareSerial.h>

#define TxPin 13                  // Tx pin 13, connect to sensor Rx pin 5
#define RxPin 12                  // Rx pin 12, connect to sensor Tx pin 7
#define BUFFSIZE 15               // used for both sensor and PC buffers

//============================================================================
SoftwareSerial GSSSerial(RxPin,TxPin); //Rx Tx
// RdPCBuffer is a linear buffer, read full command line before sending to sensor
// RdBuffer is a circular buffer, echoing characters from sensor to PC
char RdBuffer[BUFFSIZE], RdPCBuffer[BUFFSIZE];
int InS,OutS,PCPtr;               // pointers used to control the buffers

//============================================================================
void setup()
{
 pinMode(RxPin, INPUT);           // Sensor Rx
 pinMode(TxPin, OUTPUT);          // Sensor Tx
 InS=0;                           // initialise buffer pointers
 OutS=0;
 PCPtr=0;
 Serial.begin(115200);            // to USB, initialise serial via USB (baud rate can be changed)
 GSSSerial.begin(9600);           // sensor serial comms (change to 38400 for SprintIR-R & Blink)
//  GSSSerial.print("*\r\n");      // send debug command if required, to confirm serial comms
}

//============================================================================
void loop()
{
 if(ReadSensor())                 // check for data from sensor
   {
   do
    {
    Serial.print(RdBuffer[OutS]); // data available, send to terminal (PC)
    if(++OutS==BUFFSIZE)          // circular buffer, wrap pointer back to start
      OutS=0;
    }while(OutS!=InS);            // output all available chars
   }
 if(ReadPC())                     // only true when complete command received ending with'\n'
   {
```

AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

```
   RdPCBuffer[PCPtr]=0;          // null terminate string
   GSSSerial.print(RdPCBuffer); // send string received from PC to Sensor
   PCPtr=0;                      // reset string
   }
}

//============================================================================
int ReadSensor(void)
{
if(GSSSerial.available())        // have any chars been received from sensor
 {
 RdBuffer[InS++] = GSSSerial.read(); // read char
 if(InS==BUFFSIZE)               // if end of allocated buffer memory
   InS=0;                        // reset pointer, to create a circular buffer
 return true;                    // char received
 }
return false;                    // no char received
}

//============================================================================
int ReadPC(void)
{
if(Serial.available())           // has any char been received from PC
 {
 RdPCBuffer[PCPtr] = Serial.read(); // Yes, copy char to buffer
 if(RdPCBuffer[PCPtr++]=='\n')  // was it new line character
   return true;                  // full line, return true
 }
return false;                    // \n not yet received
}

//============================================================================
```

Code is also available to download here.

AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

CONCLUSION

The example software can be compiled and run using the Arduino IDE. The Arduino IDE also includes a serial monitor (Ctrl+Shift+M), which enables serial data being sent by the Arduino UNO to be analysed. The baud rate (used by the sensor) must be set using the drop-down menu on the Arduino IDE.

AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

IMPORTANT NOTICE

Gas Sensing Solutions Ltd. (GSS) products and services are sold subject to GSS's terms and conditions of sale, delivery and payment supplied at the time of order acknowledgement. GSS warrants performance of its products to the specifications in effect at the date of shipment. GSS reserves the right to make changes to its products and specifications or to discontinue any product or service without notice.

Customers should therefore obtain the latest version of relevant information from GSS to verify that the information is current. Testing and other quality control techniques are utilised to the extent GSS deems necessary to support its warranty. Specific testing of all parameters of each device is not necessarily performed unless required by law or regulation. In order to minimise risks associated with customer applications, the customer must use adequate design and operating safeguards to minimise inherent or procedural hazards. GSS is not liable for applications assistance or customer product design. The customer is solely responsible for its selection and use of GSS products. GSS is not liable for such selection or use nor for use of any circuitry other than circuitry entirely embodied in a GSS product.

GSS products are not intended for use in life support systems, appliances, nuclear systems, or systems where malfunction can reasonably be expected to result in personal injury, death or severe property or environmental damage. Any use of products by the customer for such purposes is at the customer's own risk.

GSS does not grant any licence (express or implied) under any patent right, copyright, mask work right or other intellectual property right of GSS covering or relating to any combination, machine, or process in which its products or services might be or are used. Any provision or publication of any third party's products or services does not constitute GSS's approval, licence, warranty, or endorsement thereof. Any third party trade-marks contained in this document belong to the respective third-party owner.

Reproduction of information from GSS datasheets is permissible only if reproduction is without alteration and is accompanied by all associated copyright, proprietary and other notices (including this notice) and conditions. GSS is not liable for any unauthorised alteration of such information or for any reliance placed thereon.

Any representations made, warranties given, and/or liabilities accepted by any person which differ from those contained in this datasheet or in GSS's standard terms and conditions of sale, delivery and payment are made, given and/or accepted at that person's own risk. GSS is not liable for any such representations, warranties, or liabilities or for any reliance placed thereon by any person.

ADDRESS

Gas Sensing Solutions Ltd.
Grayshill Road
Cumbernauld
G68 9HQ
United Kingdom

AN011: INTERFACING A GSS CO_2 SENSOR TO AN ARDUINO

REVISION HISTORY

DATE	RELEASE	DESCRIPTION OF CHANGES	PAGES
08/07/2021	1.0	First revision	All