

## AmbiMate MS4: Application software in Python and for Arduino



### NOTE

All numerical values are in metric units [with U.S. customary units in brackets]. Dimensions are in millimeters [and inches]. Unless otherwise specified, dimensions have a tolerance of  $\pm 0.13$  [ $\pm 0.005$ ] and angles have a tolerance of  $\pm 2^\circ$ . Figures and illustrations are for identification only and are not drawn to scale.

## 1 INTRODUCTION

### 1.1 Applications

The AmbiMate MS4 Sensor Module is suitable for sensing the characteristics of indoor environments, and utilizes a cluster of common sensors to report current conditions. The AmbiMate measures temperature, humidity, ambient light, motion, and optionally sound and/or VOC/eCO<sub>2</sub>. The AmbiMate reports data via an I2C bus to a customer supplied host PCB Assembly. The Application Software will provide an interface for both Arduino and Raspberry Pi solutions to easily read out sensor data from the AmbiMate.

### 1.2 AmbiMate Sensor Module: schematic overview

Figure 1 shows the defined sensor locations for the AmbiMate. This diagram and the terminology in Table 1 are used throughout this document to describe functionality, capabilities and design practice.

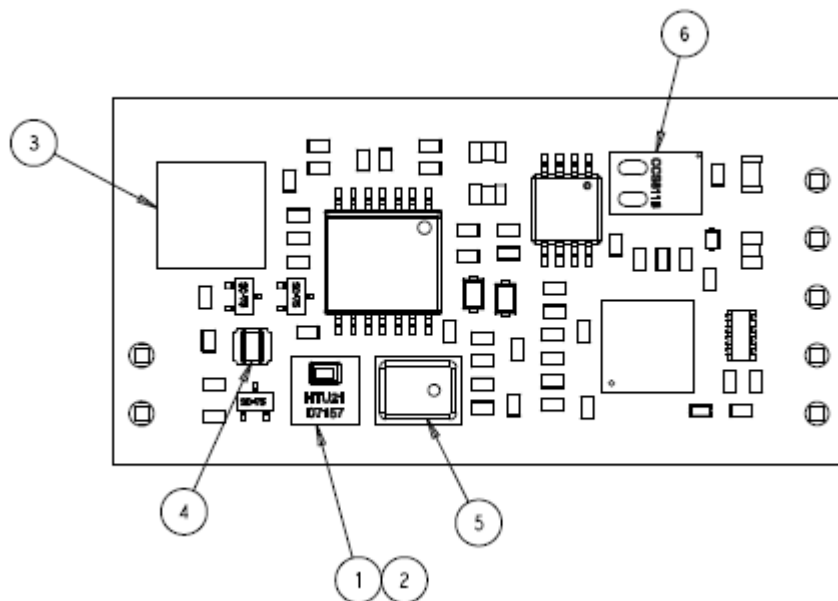


Figure 1. AmbiMate Layout

1: Temperature, 2: Humidity, 3: Motion, 4: Light, 5: Audio, 6: eCO<sub>2</sub>/VOC

### 1.3 Customer Assistance

Reference Product Base Part Number 2331211 for development kits and 2316851, 2316852, 2314277, 2314291 for the AmbiMate Sensor Module. Use of these numbers will identify the product line and help you to obtain product and tooling information when visiting [www.te.com](http://www.te.com) or calling the number at the bottom of this page.

## 2 ELECTRONIC DESIGN GUIDANCE

### 2.1 Electrical Design Guidance

| Term/Acronym     | Meaning                                                                                                                                          |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| AmbiMate         | AmbiMate Sensor Module                                                                                                                           |
| eCO <sub>2</sub> | Equivalent Carbon Dioxide                                                                                                                        |
| I <sup>2</sup> C | Inter-Integrated Circuit bus protocol. Multi-master, multi-slave, single-ended, serial computer bus                                              |
| PCBA             | Printed Circuit Board Assembly. A PCBA comprises a printed circuit board (PCB) and all the installed components to create a functional assembly. |
| PIR              | Passive Infrared                                                                                                                                 |
| SCL / SDA        | Serial Clock / Serial Data                                                                                                                       |
| VOC              | Volatile Organic Compound                                                                                                                        |

Table 1

### 2.2 Functional Element Overview

Figure 2 shows a block diagram of the functional blocks of the Application Software. The following sections overview the capability of each block. More detailed description, where applicable, can be found in the architecture and sensors sections.

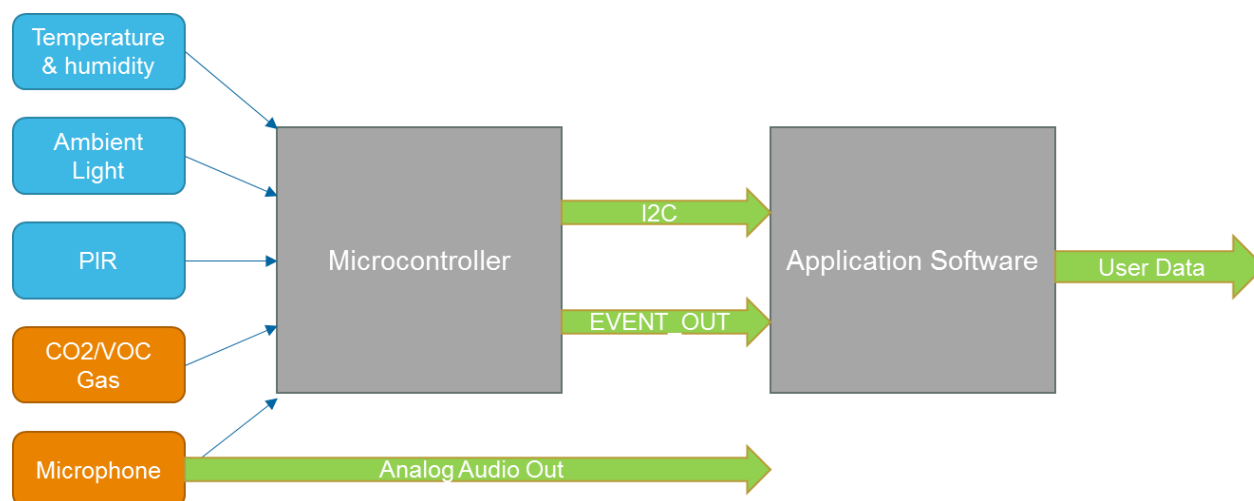


Figure 2

### 2.3 Power Supply

The AmbiMate is powered by  $3.3 \pm 0.2$  V dc required for I<sup>2</sup>C communication. This voltage need to be provided by the host board, which could be a battery or other regulated supply.

### 2.4 I<sup>2</sup>C communication

The AmbiMate is a 3.3 V dc device, therefore the I<sup>2</sup>C communication is at 3.3 V dc. This ensures compatibility with 3.3 V dc devices, but require a logic level shifter for compatibility with 5 V I<sup>2</sup>C devices. The AmbiMate ensures clock stretching, which enables the device to hold the SCL down when it transmits data. Please make sure the I<sup>2</sup>C master is capable of clock stretching to ensure valid data.

## 2.5 Compatible devices

Below is a list of compatible devices for the Application Software. Please notice that all devices capable of I<sup>2</sup>C communication can be used in combination with AmbiMate Sensor Module.

### Arduino

Due to the Arduino core (microcontroller) voltage being fixed at 5V, not all Arduino devices can be immediately connected to the AmbiMate without the usage of a logic level shifter. Below some devices that are direct compatible and devices that are not compatible without any logic level shifter.

Please note this list is not complete, therefore  $V_{IL}$  and  $V_{IH}$  levels must be checked before operating with the 3.3 V dc AmbiMate to prevent damages to the board.

#### Compatible devices:

Arduino Leonardo (Atmel ATMEGA32U4) operates at 5V, but is compatible due to LVTTTL I/O

Arduino Micro (Atmel ATMEGA32U4) operates at 5V, but is compatible due to LVTTTL I/O

Arduino Zero (Atmel ATSAMD21G18) operates at 3.3V and is therefore I/O compatible

#### Non-compatible devices:

Arduino Mega2560 (Atmel ATMEGA2560)

Arduino Uno (Atmel ATMEGA328P)

Arduino Nano (Atmel ATMEGA328P)

#### Connection

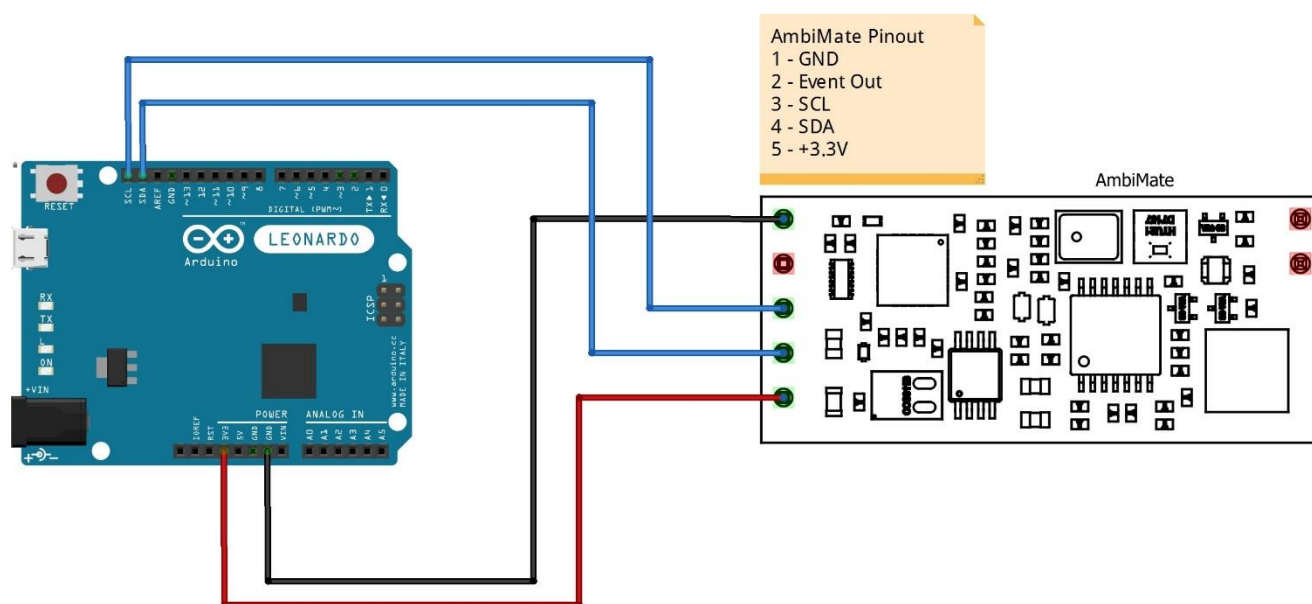


Figure 3: Connection diagram for Arduino

## Raspberry Pi

All Raspberry Pi products are compatible with the Application Software and the AmbiMate.

### Baud rate

For connection with the Raspberry Pi, the baud rate must be slowed down in order to receive correct data. The recent microcontroller on a Raspberry Pi, Broadcom BCM2835 ARM processor does not support clock stretching which is necessary when the AmbiMate pulls down the clock. When the AmbiMate holds down the clock from the master to make sure all data is valid and clocked, the Pi does not ensure the next clock does have a valid length, therefore data is not read correctly.

The default baud rate of I<sup>2</sup>C is 100kbaud. This can be slowed down to 10kbaud to ensure data is valid. This will not have any influences on the functionality of the AmbiMate. Please refer to Chapter 5 Raspberry Pi: Baud Rate Change for a description how to slow down the clock speed.

### Connection

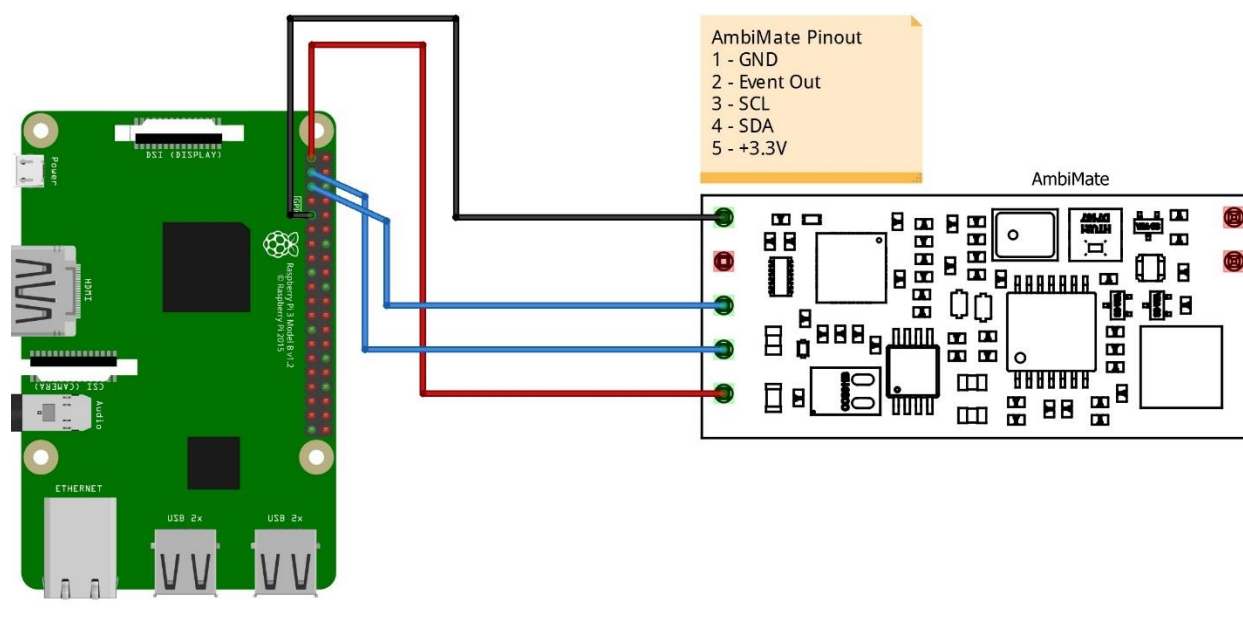


Figure 4: Connection diagram for Raspberry Pi

## 2.6 Pin layout

The top view pin layout is shown in Figure 5.

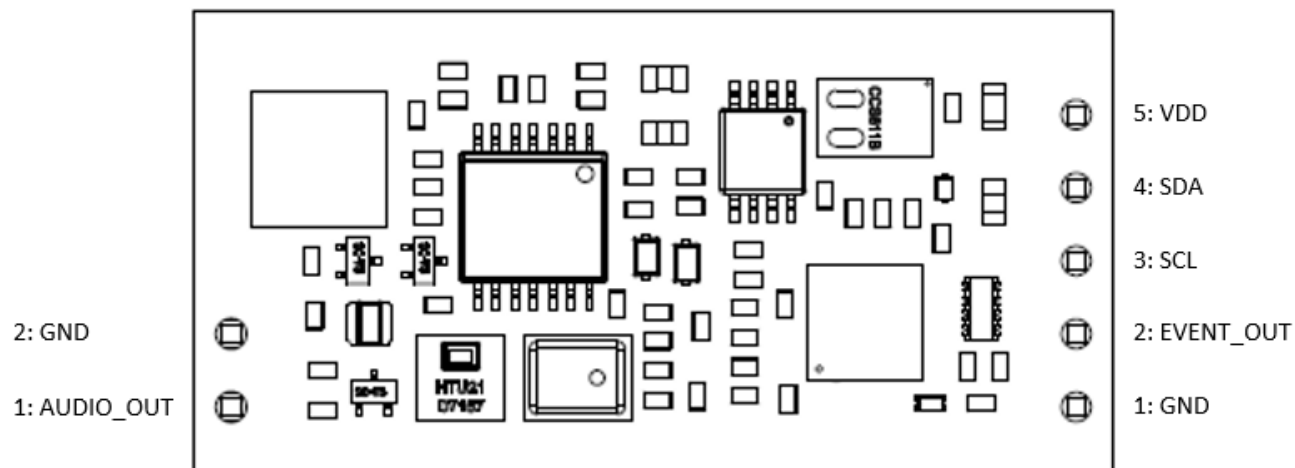


Figure 5

## 2.7 Registers

The following tables describes the readable and writeable registers of the AmbiMate.

| Register Address | Data Value                 |
|------------------|----------------------------|
| 0x00             | Status High Byte           |
| 0x01             | Temperature High Byte      |
| 0x02             | Temperature Low Byte       |
| 0x03             | Humidity High Byte         |
| 0x04             | Humidity Low Byte          |
| 0x05             | Light High Byte            |
| 0x06             | Light Low Byte             |
| 0x07             | Audio High Byte            |
| 0x08             | Audio Low Byte             |
| 0x09             | Battery Volts High Byte    |
| 0x0A             | Battery Volts Low Byte     |
| 0x0B             | eCO <sub>2</sub> High Byte |
| 0x0C             | eCO <sub>2</sub> Low Byte  |
| 0x0D             | VOC High Byte              |
| 0x0E             | VOC Low Byte               |

*Sensor data registers.*

| Register Address | Data Value                 |
|------------------|----------------------------|
| 0x40             | Status High Byte           |
| 0x41             | Temperature High Byte      |
| 0x42             | Humidity High Byte         |
| 0x43             | Light High Byte            |
| 0x44             | Audio High Byte            |
| 0x45             | Battery Volts High Byte    |
| 0x46             | eCO <sub>2</sub> High Byte |
| 0x47             | VOC High Byte              |

*8-bit High Byte Sensor data registers.*

| Register Address | Data Value           |
|------------------|----------------------|
| 0x80             | Firmware version     |
| 0x81             | Firmware sub-version |
| 0x82             | Optional Sensors     |

*Other registers.*

| Register Address | Data Value      |
|------------------|-----------------|
| 0xC0             | Scan Start Byte |

*Writeable register*

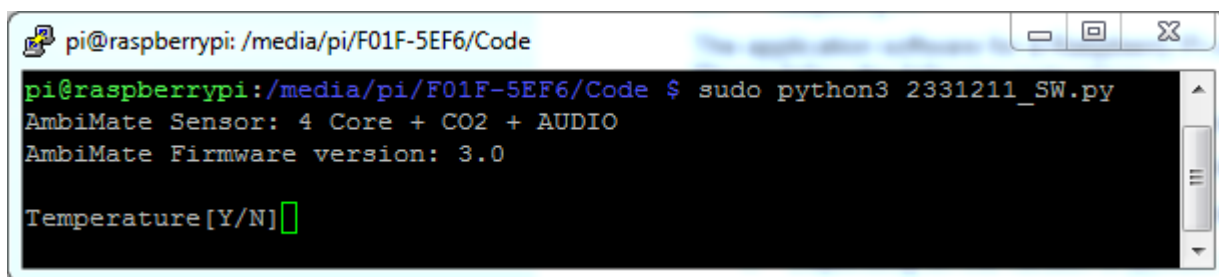
### 3 APPLICATION SOFTWARE GUIDANCE

The software provides functionality to read sensor data of the AmbiMate. Data sets can be chosen by a line command and sample rate of reading can be set. Application Software can be downloaded at [www.te.com/AmbiMate](http://www.te.com/AmbiMate)

#### 3.1 Raspberry Pi

The application software for a Raspberry Pi contains a python file which can be used on a Raspberry Pi. Please follow the following instructions:

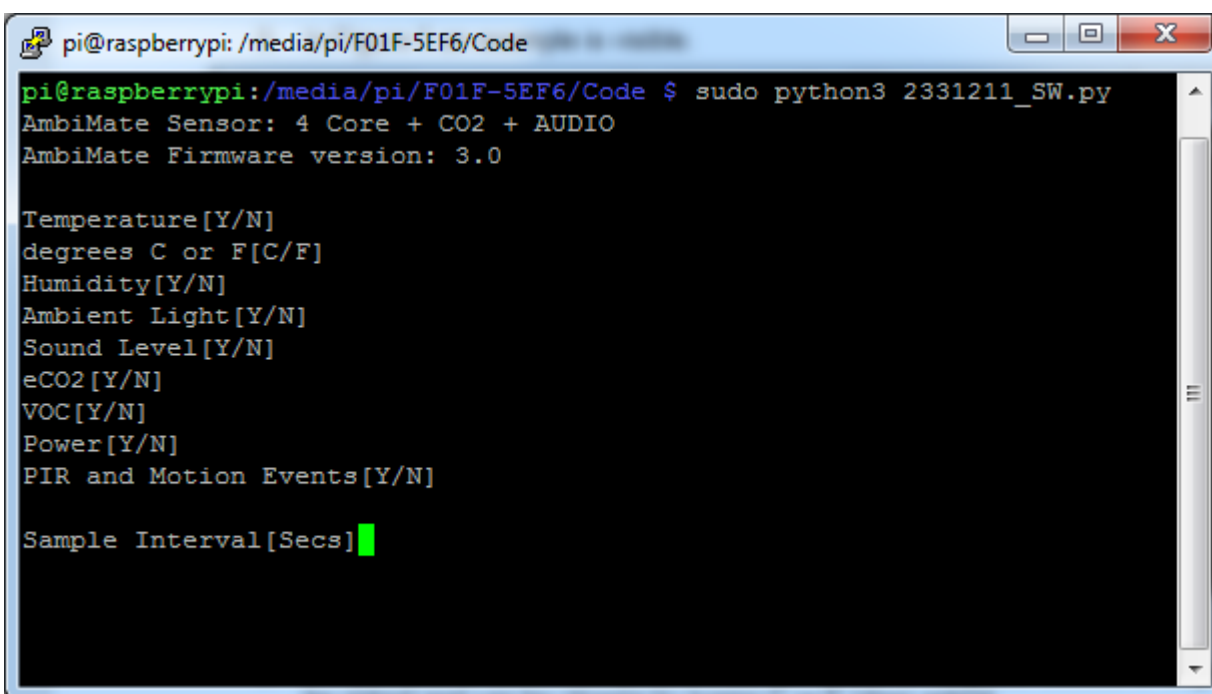
1. Set up the Pi with settings applicable for the Raspberry Pi
2. Make sure the I2C is enabled for using the smbus(1) (1 for Pi versions V2 and V3)
3. Run the program in python3 or python (python 2 is standard)
4. Depending on the version of python, if python2 is used, please type 'y' brackets around the serial input
5. If python3 is used, brackets around an answer are no longer needed
6. Data will be visible as shown in Figure 6.



```
pi@raspberrypi: /media/pi/F01F-5EF6/Code
pi@raspberrypi:/media/pi/F01F-5EF6/Code $ sudo python3 2331211_SW.py
AmbiMate Sensor: 4 Core + CO2 + AUDIO
AmbiMate Firmware version: 3.0
Temperature[Y/N]
```

Figure 6: Initial data

7. By entering Y or N, data sets can be chosen. Therefore, only sensor data which is necessary is visible and by typing a N, not wanted data will be left out of the set. Only the sensors mounted and present on the board will be asked for data. If a sensor is not listed as present, it will not be visible.
8. In Figure 7: Example of sensor set an example is visible.



```
pi@raspberrypi: /media/pi/F01F-5EF6/Code
pi@raspberrypi:/media/pi/F01F-5EF6/Code $ sudo python3 2331211_SW.py
AmbiMate Sensor: 4 Core + CO2 + AUDIO
AmbiMate Firmware version: 3.0
Temperature[Y/N]
degrees C or F[C/F]
Humidity[Y/N]
Ambient Light[Y/N]
Sound Level[Y/N]
eCO2 [Y/N]
VOC[Y/N]
Power[Y/N]
PIR and Motion Events[Y/N]
Sample Interval[Secs]
```

Figure 7: Example of sensor set

9. It is possible to set the temperature sensor output on degrees Celsius or degrees Fahrenheit. This will be asked and can be chosen by typing C or F when asked.
10. The minimum possible sample rate is 0.5 seconds.
11. Please consider the VOC/eCO<sub>2</sub> sensor does have an internal sample rate limited at 60 seconds.
12. Data will be shown as visible in the following figure.

```
pi@raspberrypi:/media/pi/F01F-5EF6/Code $ sudo python3 2331211_SW.py
AmbiMate Sensor: 4 Core + CO2 + AUDIO
AmbiMate Firmware version: 3.0

Temperature[Y/N]y
degrees C or F[C/F]C
Humidity[Y/N]y
Ambient Light[Y/N]y
Sound Level[Y/N]y
eCO2[Y/N]y
VOC[Y/N]y
Power[Y/N]y
PIR and Motion Events[Y/N]y

Sample Interval[Secs]1

Temp (C)    Humidity    Light (Lux)    Sound (dB)    eCO2 (PPM)    VOC (PPB)    Power (volts)    Events
24.7        37.1        698            20            400           0            3.36             -
24.7        36.9        722            58            400           0            3.36             -
24.7        36.9        723            20            400           0            3.36             -
24.7        36.9        722            58            400           0            3.36             -
24.7        36.9        724            64            400           0            3.36             -
```



## 3.2 Arduino

The application software for Arduino contains a .ino file which can be uploaded using the Arduino IDE. Please follow the following instructions:

1. Load the .ino file in Arduino IDE.
2. Choose the serial output port via: Tools > Port > COM...
3. Open the Serial Monitor before uploading the sketch on to the board
4. Make sure the Serial Monitor is set at 9600 baud with No Line Ending
5. After uploading to the board, the Serial Monitor shows the data as in Figure 9. Firmware version and sensor availability can be checked.

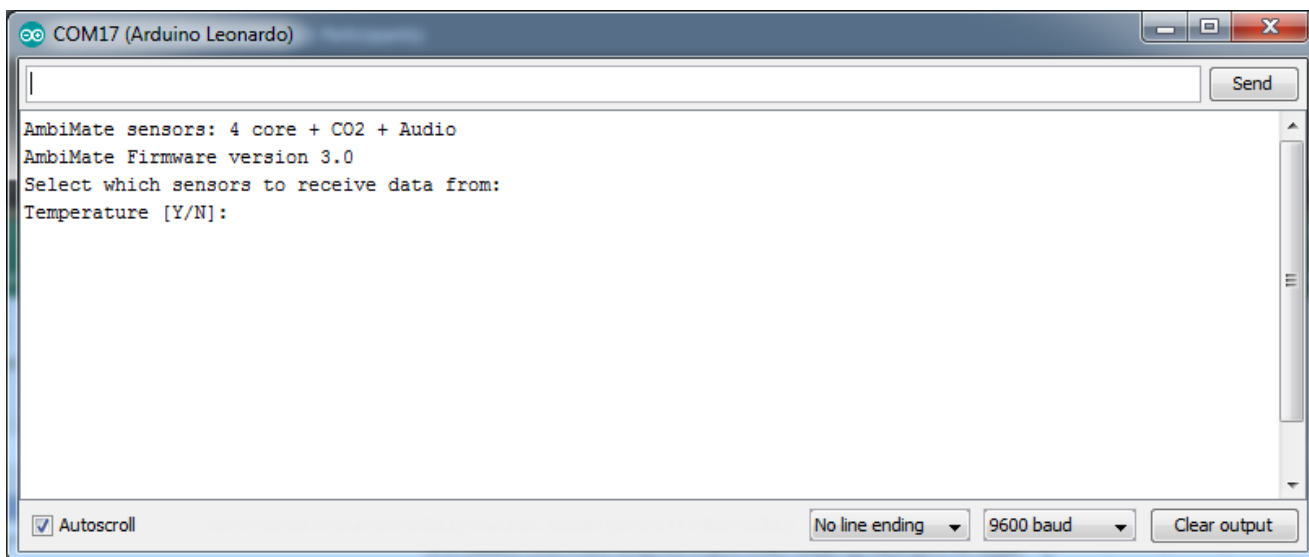


Figure 9

6. By entering Y or N, data sets can be chosen.
7. The following sensor sets can be set:

```

Select which sensors to receive data from:
Temperature [Y/N]:
deg C or F [C/F]:
Humidity [Y/N]:
Ambient Light [Y/N]:
Audio [Y/N]:
eCO2 [Y/N]:
VOC [Y/N]:
Power [Y/N]:
PIR and Motion Events [Y/N]:

Sample Interval[Secs]:
  
```

8. It is possible to set the temperature sensor output on degrees Celsius or degrees Fahrenheit.
9. The minimum possible sample rate is 0.5 seconds.
10. New data sets can be chosen by resetting the program by entering 'R'. The program will start again with the sensor selection.

11. Please consider the eCO2 sensor does have an internal sample rate limited at 60 seconds.
12. Output is visible as shown in below figure.

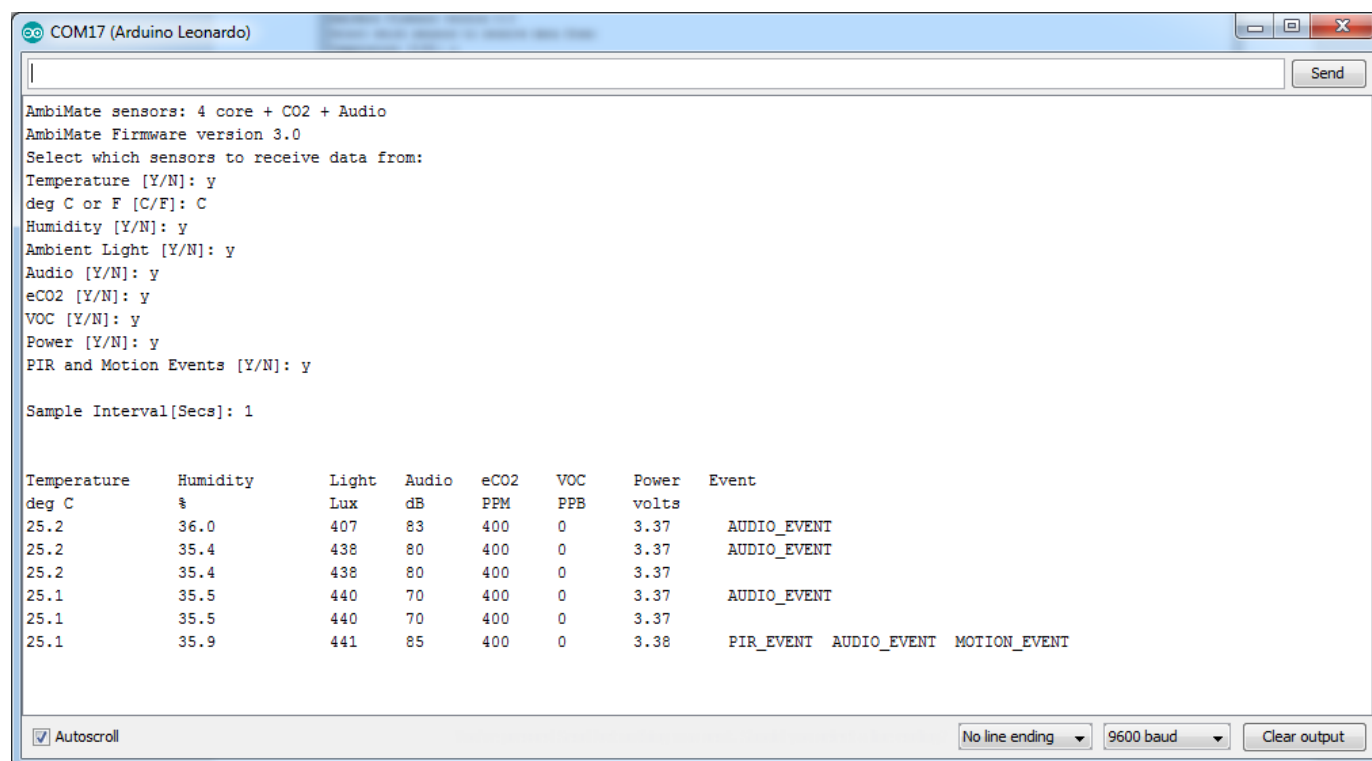


Figure 10

## 4 SOFTWARE DESIGN GUIDANCE

Both Arduino and Python source codes can be used as example for firmware writing. The following details are important for designing firmware for the AmbiMate.

1. I2C address of the AmbiMate is fixed at 0x2A
2. Before every read out, register 0xC0 must be initialized to scan the desired register set. 0xFF can be send to scan the complete register set.
3. Data of Temperature and Humidity have a 0.1-degree/percent resolution
4. Data of Light, Audio, eCO<sub>2</sub> and VOC are in Lux, Decibels, PPM and PPB.
5. Data of Battery Voltage uses a conversion:  $(\text{Data} / 1024.0) * (3.3 / 0.330)$
6. eCO<sub>2</sub>/VOC sensor does have an internal sample rate of 60 seconds
7. All other sensors do have a selectable sample rate from 0.5 seconds or more
8. Registers 0x00 to 0x0E provide sensor data on a detailed level. The high byte data set at registers 0x40 to 0x47 can be used to implement if detailed levels are not required. Please notice the high byte only contains data in steps of dec. 256, 0x0100.

## 5 RASPBERRY PI: BAUD RATE CHANGE

As described in Chapter 2.5, the Raspberry Pi clock speed must be slowed down to ensure valid data.

*Via SD card on PC*

To reduce the rate, shutdown the Pi and use the SD card in a PC to search for the config file on the SD card [config.txt](#). Open this with a text editor and insert underneath hardware interfaces the following:  
[dtparam=i2c\\_baudrate=10000](#)

Save the file and install the SD card back in the Pi for this updated baud rate.

*Via command terminal on Pi*

To reduce the rate, go to [/boot/config.txt](#) and open using *cat* or *nano*. Scroll down to the section of hardware interfaces and insert the following: [dtparam=i2c\\_baudrate=10000](#)

Save the file and shutdown the Pi using [sudo reboot](#).

```
# Uncomment some or all of these to enable the optional hardware
interfaces
dtparam=i2c_arm=on
dtparam=i2c_baudrate=10000
dtparam=i2s=on
#dtparam=spi=on
```

Add or Edit to look like this

Figure 11: I2C baudrate command