

Quick Tutorials: B-CAM-M8C-00



B-CAM-M8C-00 is a full-featured breakout test and development board for the u-blox CAM-M8C GPS/GNSS module. Much effort has been given to lay the board out to the manufacturer's specifications, meeting all ground plane recommendations.

The datasheet can be found at petrichorlabs.io/docs/datasheet_B-CAM-M8C-00.pdf.

Many variations of the tutorials given here could be given instead. We give two examples of how the NMEA data can be accessed from the board. Both SPI and I2C can be used via a USB to serial converter or via an Arduino Uno, or some similar board.

For details on using the board's V_BCKP or external antenna functionality, see the datasheet.

1. Connect with SPI via a USB to Serial Converter 1
2. Connect with I2C via an Arduino Uno 5

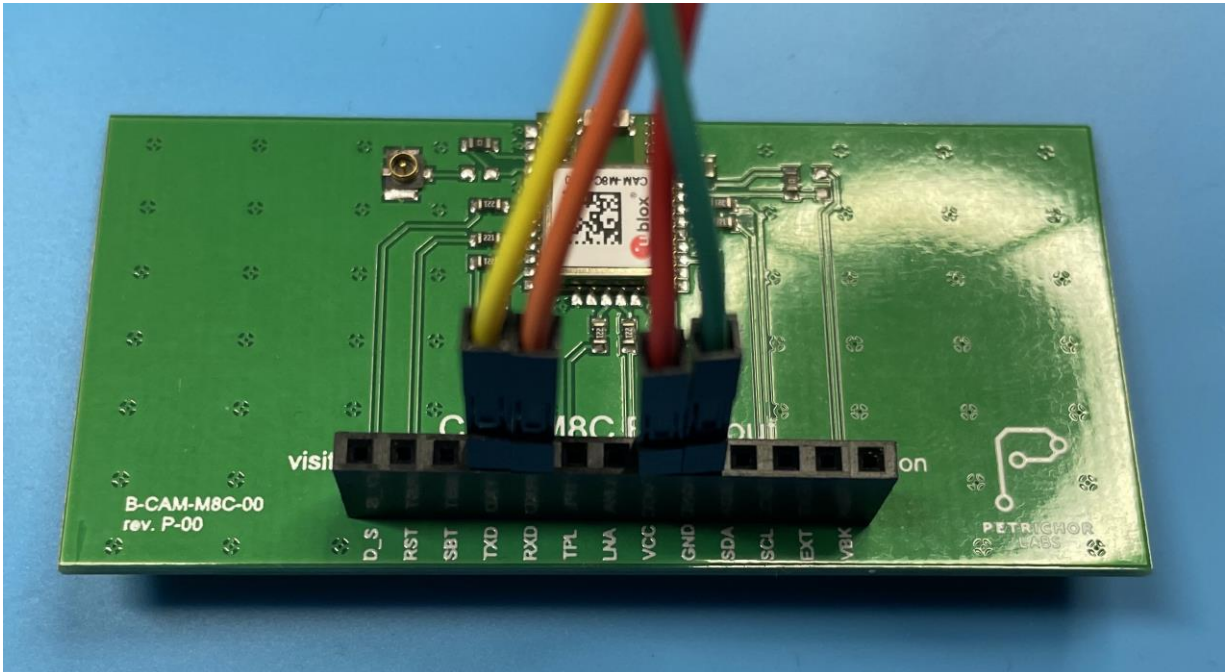
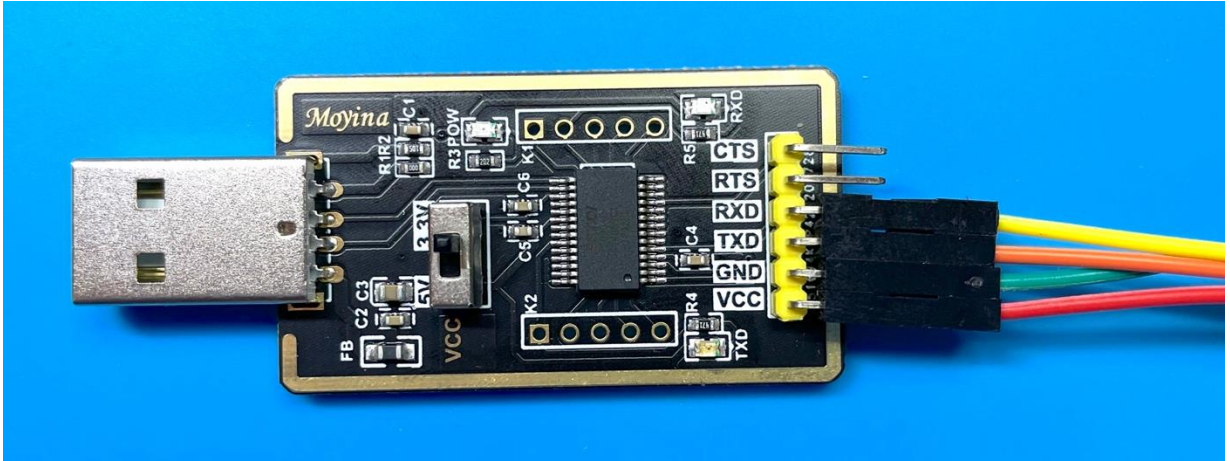
1. Connect with SPI via a USB to Serial Converter

1. Obtain a USB to serial converter device and female to male jumper wires. If you use a converter device like [this one](#) (shown in the first photo below), you should be aware of the issue and the solution to that issue described [here](#). We were able to use [this one](#) (shown in the second photo below) without any driver issues.
2. Connect the breakout board to the serial converter with the following pin mapping and as shown in the photos below.

Serial Converter		Breakout Board
3V3	↔	VCC
TXD	↔	RXD
RXD	↔	TXD
GND	↔	GND
all other pins		do not connect



or



3. Connect the USB device to your computer and identify the USB to Serial converter device in your operating system.

In **macOS or Linux**, this can be done with the following process on a command line. See the first screenshot in Step 5 for an example.

- a. `ls /dev > ~/Desktop/dev_list_without`
- b. [plug USB device into computer]
- c. `ls /dev > ~/Desktop/dev_list_with`
- d. `diff ~/Desktop/dev_list_with ~/Desktop/dev_list_without`
- e. `rm ~/Desktop/dev_list_with*`
- f. The diff should show you the system designation of the serial device. If you see a 'cu' and a 'tty' device, use the 'cu' device.

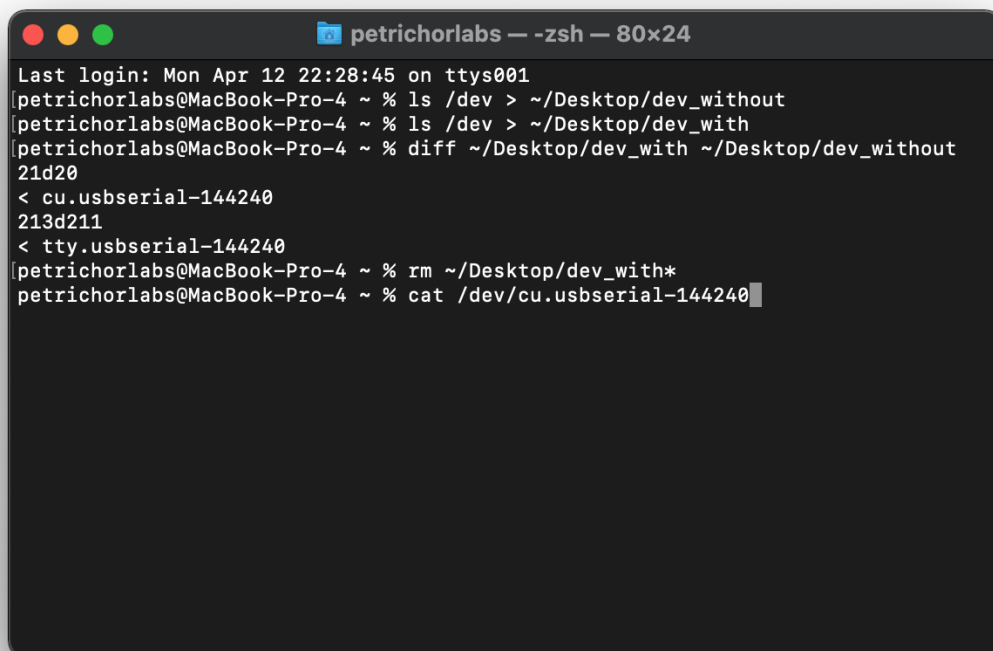
In **Windows**, this can be done with the following process on a command line like PowerShell or Command Prompt:

- a. `mode`
- b. [plug USB device into computer]
- c. `mode`

The COMX device that was listed in (c) but not in (a) is what should use in Step 5.

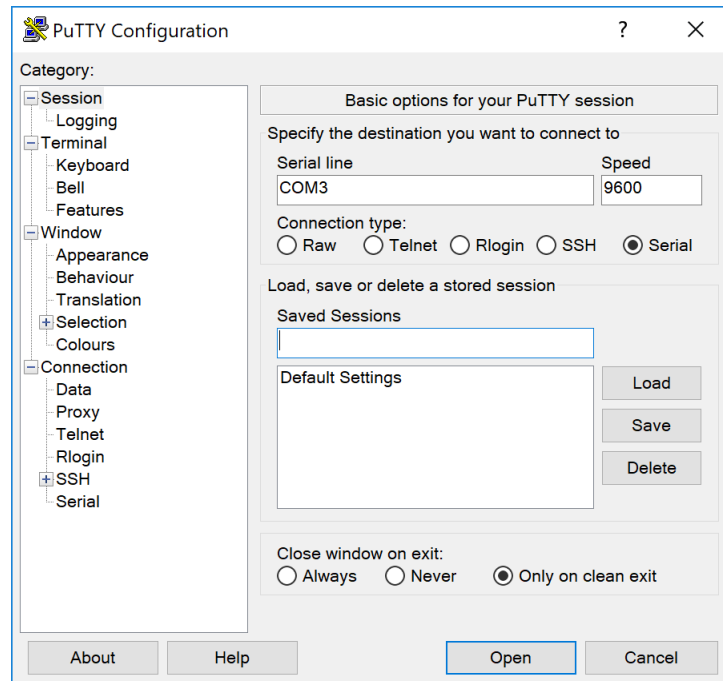
4. View the serial data coming from the CAM-M8C module.

In **macOS or Linux**, this can be done by calling 'cat' for the device:



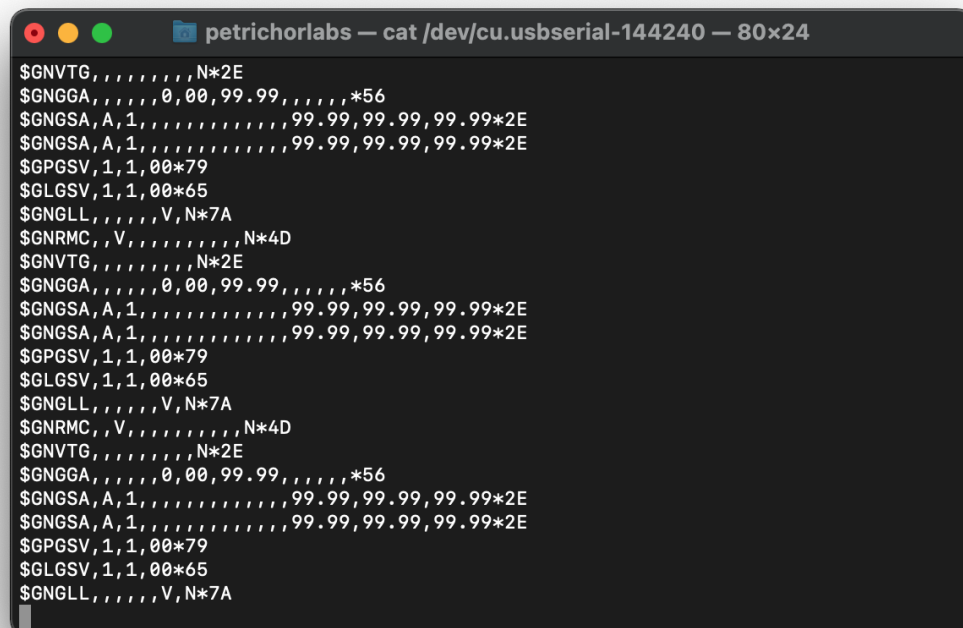
```
petrichorlabs — zsh — 80x24
Last login: Mon Apr 12 22:28:45 on ttys001
petrichorlabs@MacBook-Pro-4 ~ % ls /dev > ~/Desktop/dev_without
petrichorlabs@MacBook-Pro-4 ~ % ls /dev > ~/Desktop/dev_with
petrichorlabs@MacBook-Pro-4 ~ % diff ~/Desktop/dev_with ~/Desktop/dev_without
21d20
< cu.usbserial-144240
213d211
< tty.usbserial-144240
petrichorlabs@MacBook-Pro-4 ~ % rm ~/Desktop/dev_with*
petrichorlabs@MacBook-Pro-4 ~ % cat /dev/cu.usbserial-144240
```

In **Windows**, you can setup a connection in a program like PuTTY. Be sure to set the speed to 9600:



And then click “Open”.

You should begin to see data from the module:



```

petrichorlabs — cat /dev/cu.usbserial-144210 — 80x24
$GLGSV,2,1,07,65,32,225,,71,45,032,31,72,78,253,22,85,04,141,*61
$GLGSV,2,2,07,86,51,134,16,87,55,333,38,88,15,324,31*5D
$GNLL,3933.87828,N,10506.29957,W,044724.00,A,A*61
$GNRMC,044725.00,A,3933.87809,N,10506.29949,W,0.286,,130421,,A*7C
$GNVTG,,T,M,0.286,N,0.530,K,A*37
$GNGGA,044725.00,3933.87809,N,10506.29949,W,1,07,5.19,1745.5,M,-21.7,M,,*48
$GNGSA,A,3,18,20,27,13,,,,,,,,,6.37,5.19,3.69*13
$GNGSA,A,3,87,71,88,,,,,,,,,6.37,5.19,3.69*16
$GPGSV,3,1,09,05,04,066,,10,37,262,08,13,24,045,26,18,77,032,31*73
$GPGSV,3,2,09,20,66,034,36,26,05,257,,27,20,318,30,29,30,172,*72
$GPGSV,3,3,09,32,04,204,*43
$GLGSV,2,1,07,65,32,225,,71,45,032,31,72,78,253,21,85,04,141,*62
$GLGSV,2,2,07,86,51,134,16,87,55,333,38,88,15,324,31*5D
$GNLL,3933.87809,N,10506.29949,W,044725.00,A,A*6C
$GNRMC,044726.00,A,3933.87771,N,10506.29941,W,0.021,,130421,,A*78
$GNVTG,,T,M,0.021,N,0.039,K,A*34
$GNGGA,044726.00,3933.87771,N,10506.29941,W,1,07,5.19,1745.0,M,-21.7,M,,*46
$GNGSA,A,3,18,20,27,13,,,,,,,,,6.37,5.19,3.69*13
$GNGSA,A,3,87,71,88,,,,,,,,,6.37,5.19,3.69*16
$GPGSV,3,1,09,05,04,066,,10,37,262,09,13,24,045,26,18,77,032,31*72
$GPGSV,3,2,09,20,66,034,36,26,05,257,,27,20,318,30,29,30,172,*72
$GPGSV,3,3,09,32,04,204,21*40
$GLGSV,2,1,07,65,32,226,,71,45,032,31,72,78,253,21,85,04,141,*61
$GLGSV,2,2,0

```

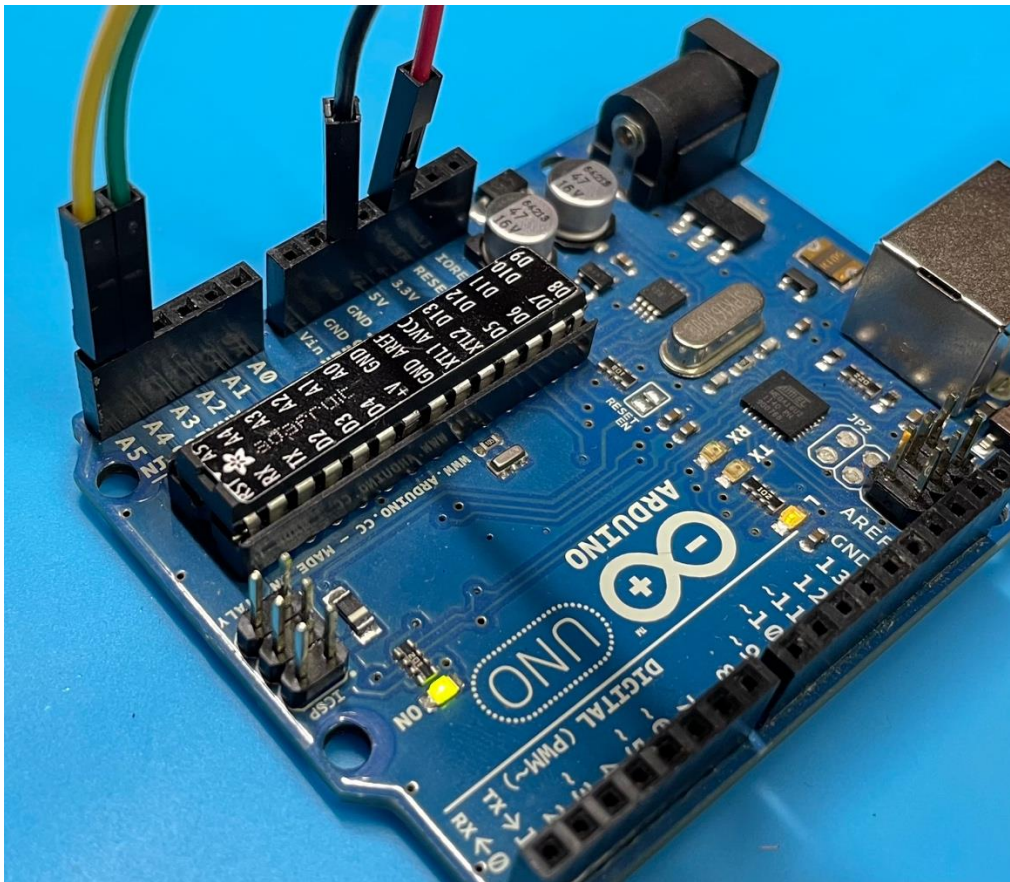
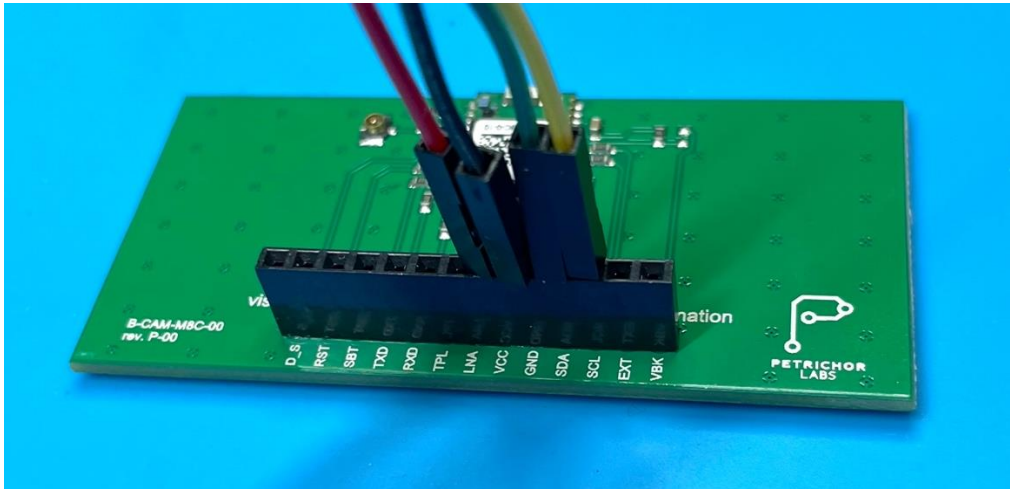
After several minutes, if you do not have a fix so that the data looks like the last screenshot above, consider moving the breakout board near a window or outdoors.

2. Connect with I2C via an Arduino Uno

1. Obtain an Arduino Uno and female to male jumper wires.
2. Connect the breakout board to the Arduino with the following pin mapping and as shown in the photos below.

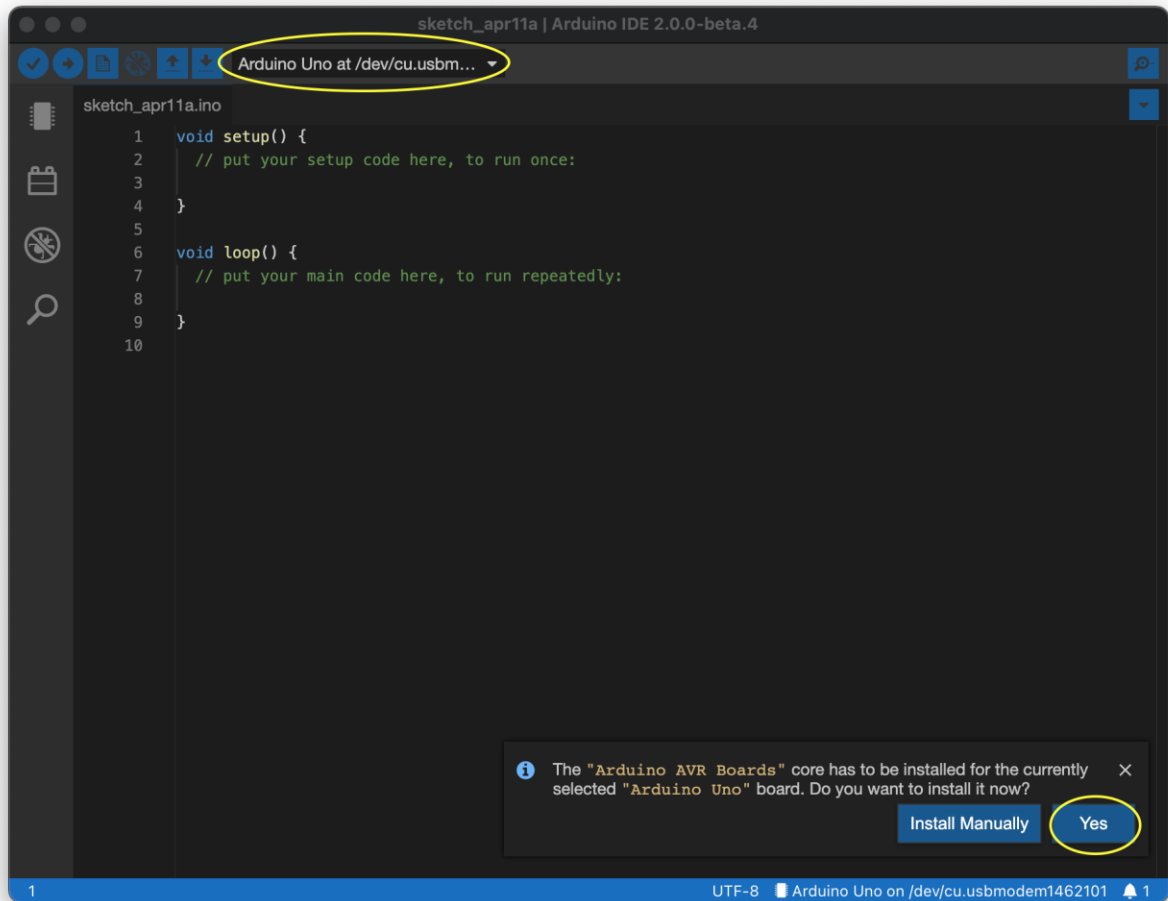
Arduino Uno		Breakout Board
3.3V	↔	VCC
GND	↔	GND
analog pin A4 (SDA)	↔	SDA
analog pin A5 (SCL)	↔	SCL

Reference [this page](#) for help finding the SCL and SDA pins on the Arduino.

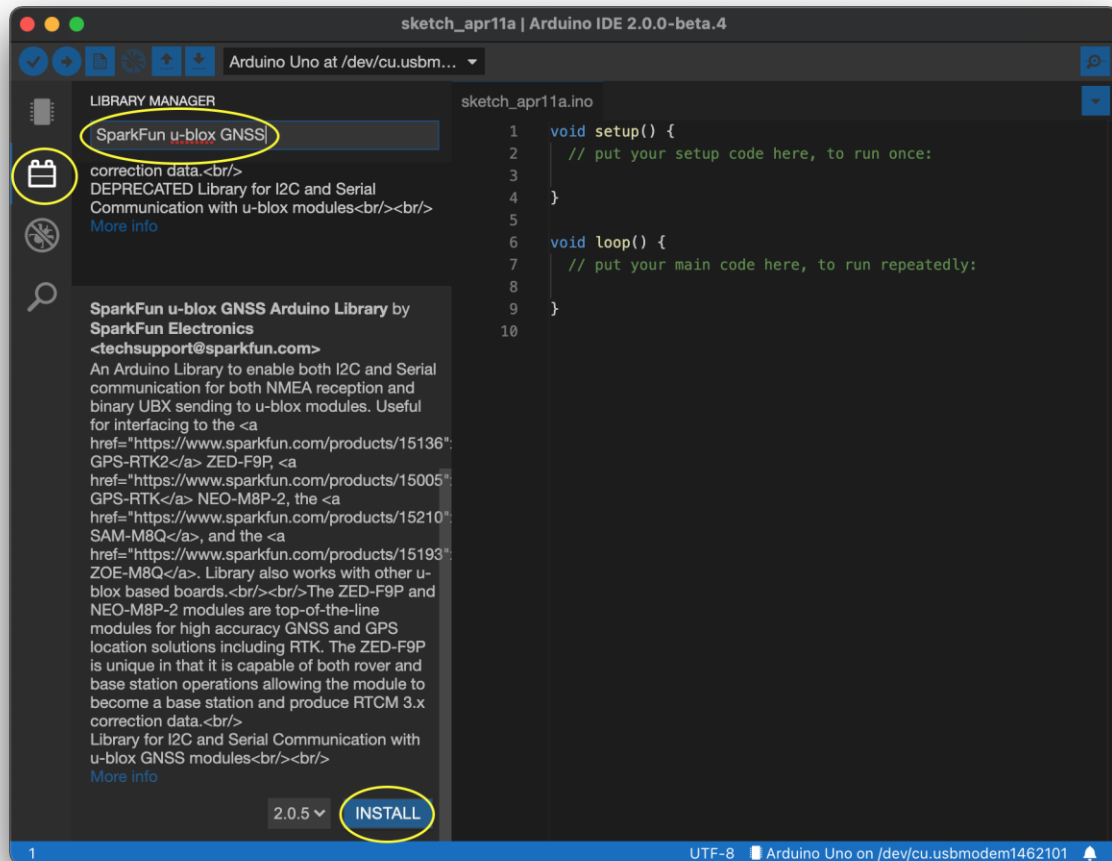


3. Connect the Arduino Uno to your computer.
4. Open the Arduino IDE. Version 2.0.0-beta.4 is depicted below.

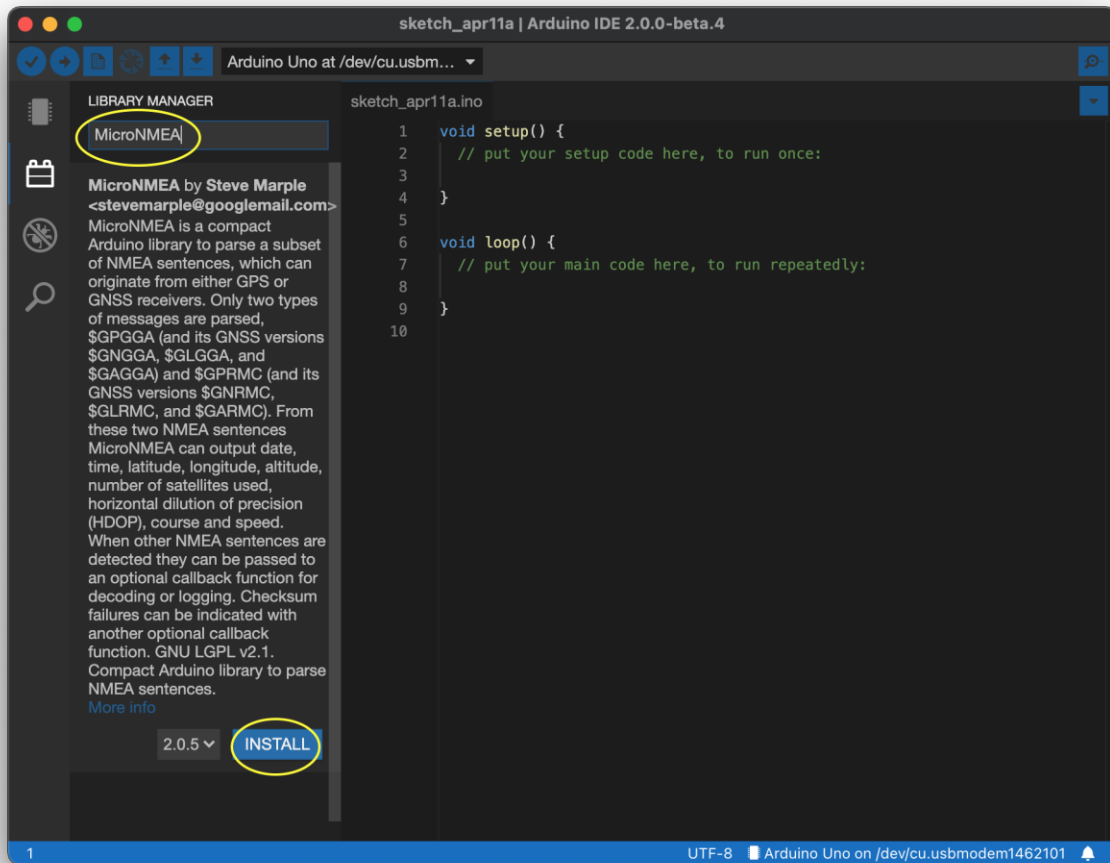
5. Select your Arduino from the menu at the top. If you haven't previously, install the board support files for the Arduino.



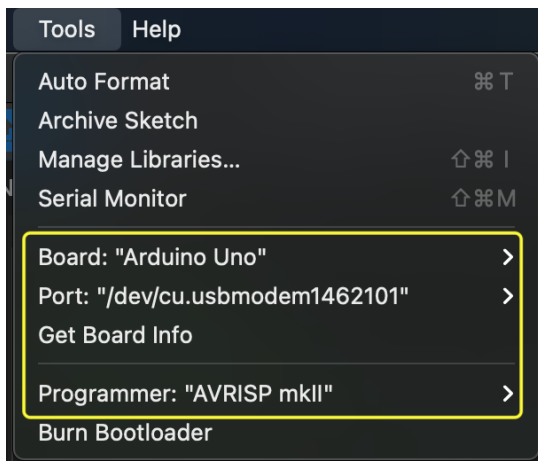
6. Select the Library Manager tab. Search for "SparkFun u-blox GNSS" and find "SparkFun u-blox GNSS Arduino Library", not the deprecated "SparkFun u-blox Arduino Library," in the list. Select the latest version and click "Install."



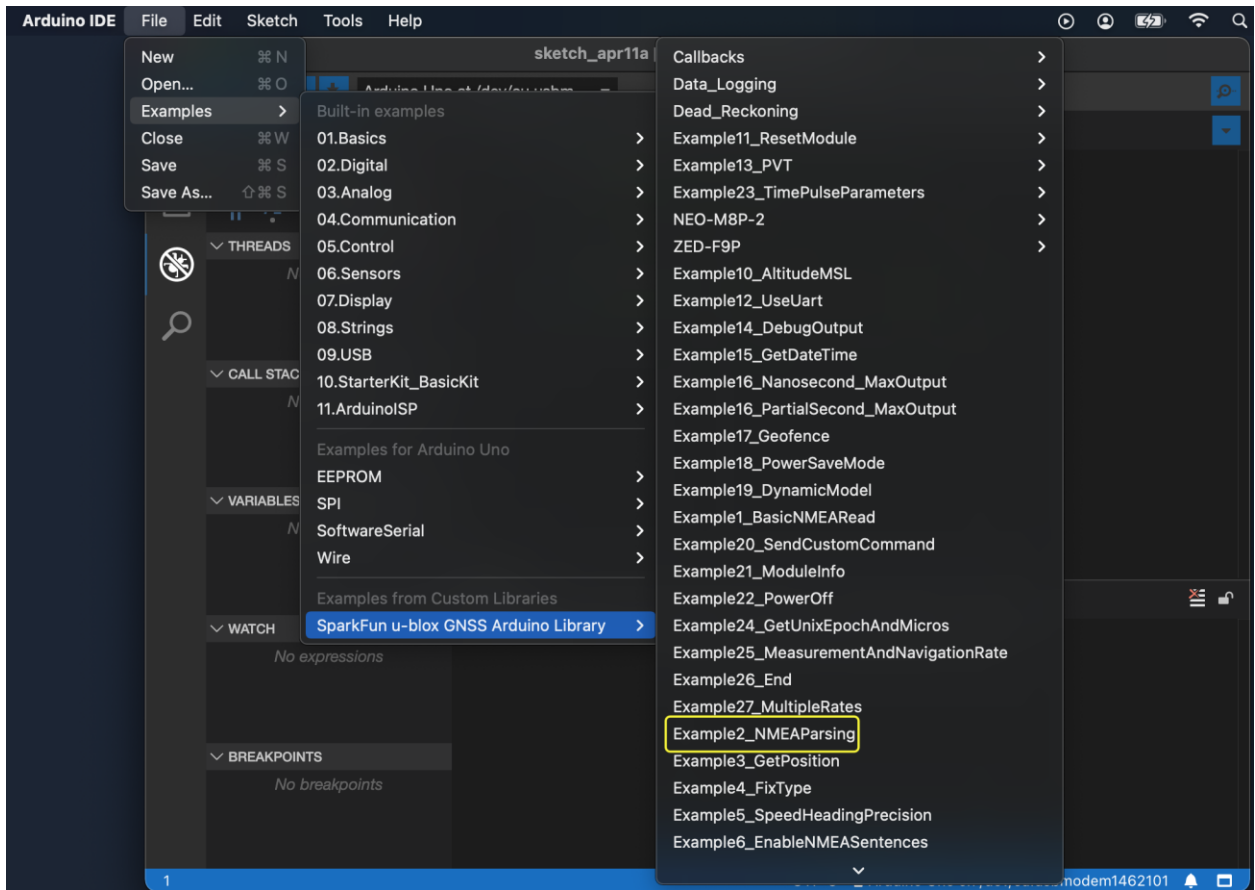
7. Search for "MicroNMEA" and find "MicroNMEA by Steve Marple" in the list. Select the latest version and click "Install."



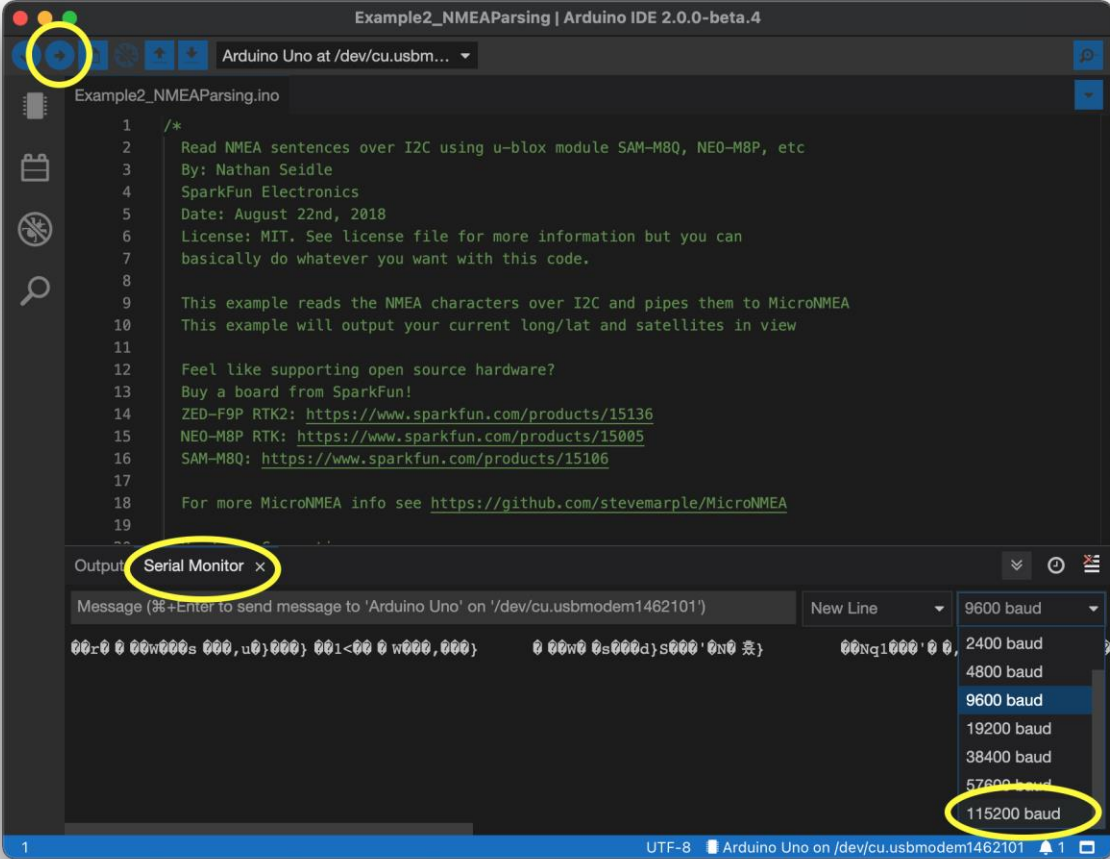
8. Make sure you have the correct board, port, and programmer settings.



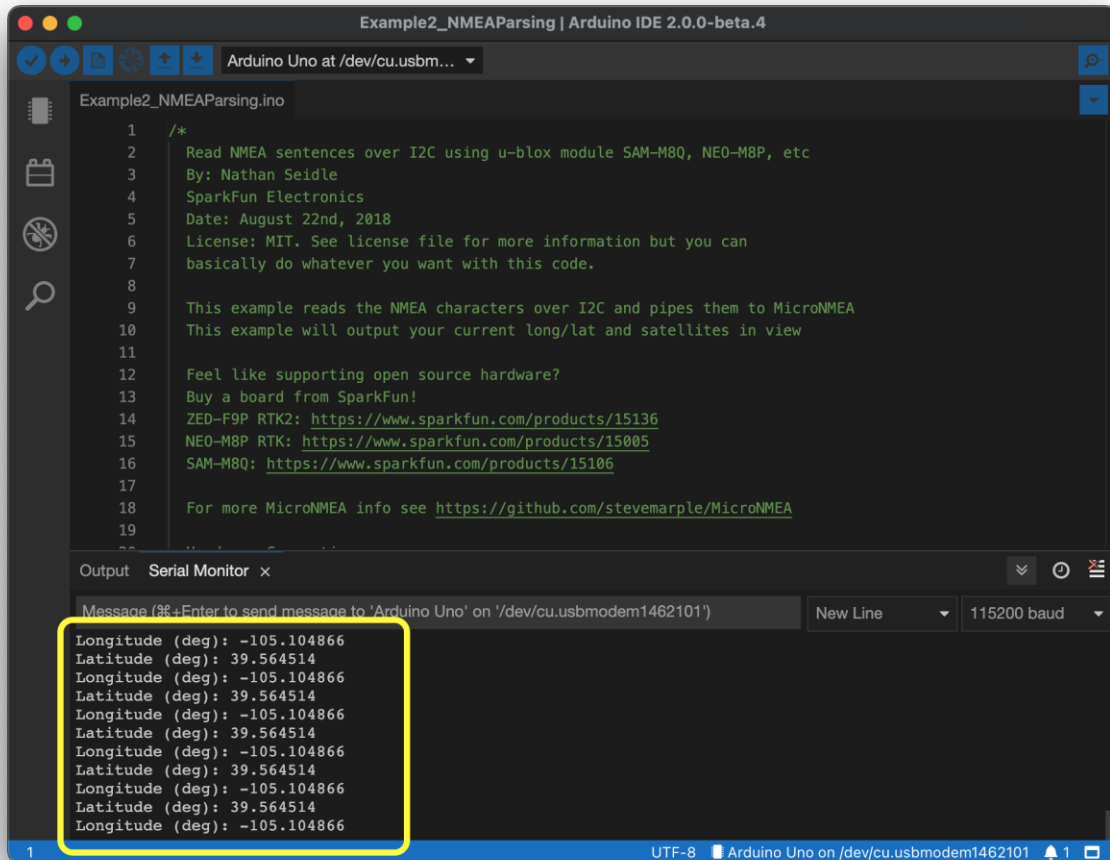
- Open the u-blox NMEAParsing example by going to Examples > SparkFun u-blox GNSS Arduino Library > Example2_NMEAParsing.



10. Verify and upload the code to the Arduino. Open the Serial Monitor (on macOS: Tools > Serial Monitor). Set the baud rate for the Serial Monitor to 115200.



11. You should begin to see parsed data from the module:



The screenshot shows the Arduino IDE interface. The top window displays the code for 'Example2_NMEAParsing.ino'. The code includes a header comment with author information (Nathan Seidle, SparkFun Electronics, August 22nd, 2018) and a license (MIT). It describes the example as reading NMEA characters over I2C and outputting current longitude, latitude, and satellite information. It also provides links for purchasing SparkFun boards (ZED-F9P RTK2, NEO-M8P RTK, SAM-M8Q) and a GitHub repository for MicroNMEA.

The bottom window is the Serial Monitor, which shows the following output:

```
Longitude (deg): -105.104866  
Latitude (deg): 39.564514  
Longitude (deg): -105.104866  
Latitude (deg): 39.564514  
Longitude (deg): -105.104866  
Latitude (deg): 39.564514  
Longitude (deg): -105.104866  
Latitude (deg): 39.564514  
Longitude (deg): -105.104866  
Latitude (deg): 39.564514  
Longitude (deg): -105.104866
```

The serial monitor settings are set to 'New Line' and '115200 baud'.

After several minutes, if you do not have a fix so that the data looks like the last screenshot above, consider moving the breakout board near a window or outdoors.

12. Experiment with the other u-blox example files. Example1 will give you the raw NMEA sentences:

The screenshot shows the Arduino IDE interface. The top window displays the source code for 'Example1_BasicNMEARead.ino'. The code includes a header with a license and a description of the program's purpose: reading NMEA sentences from a u-blox module over I2C and outputting them to the serial port. It also provides links to SparkFun products: ZED-F9P RTK, NEO-M8P RTK, and SAM-M8Q. The bottom window is the Serial Monitor, which shows the raw NMEA sentences received from the module. The sentences are highlighted with a yellow box.

```
Example1_BasicNMEARead.ino
1  /*
2  Read NMEA sentences over I2C using u-blox module SAM-M8Q, NEO-M8P, ZED-F9P, etc
3  By: Nathan Seidle
4  SparkFun Electronics
5  Date: August 22nd, 2018
6  License: MIT. See license file for more information but you can
7  basically do whatever you want with this code.
8
9  This example reads the NMEA sentences from the u-blox module over I2c and outputs
10 them to the serial port
11
12 Feel like supporting open source hardware?
13 Buy a board from SparkFun!
14 ZED-F9P RTK2: https://www.sparkfun.com/products/15136
15 NEO-M8P RTK: https://www.sparkfun.com/products/15005
16 SAM-M8Q: https://www.sparkfun.com/products/15106

```

```
Serial Monitor
Message (%+Enter to send message to 'Arduino Uno' on '/dev/cu.usbmodem1462101')
New Line 115200 baud
$GNGSA,A,3,03,02,24,17,28,06,12,19,,,,,1.63,0.95,1.32*12
$GNGSA,A,3,82,80,81,73,,,,,,1.63,0.95,1.32*1B
$GPGSV,3,1,09,02,56,212,31,03,10,039,25,06,80,097,27,12,37,312,29*76
$GPGSV,3,2,09,14,14,127,11,17,33,063,28,19,54,046,21,24,39,255,25*7D
$GPGSV,3,3,09,28,23,123,23*4A
$GLGSV,3,1,10,66,07,052,18,67,02,098,,73,48,318,19,74,02,332,*64
$GLGSV,3,2,10,79,16,159,,80,69,193,17,81,07,046,19,82,51,022,40*65
$GLGSV,3,3,10,83,62,276,,84,10,237,*63
$GNGLL,3933.87071,N,10506.29723,W,220616.00,A,A*68
$GNRMC,220617.00,A,3933.87070,N,10506.29724,W,0.078,,110421,,A*7E
$GNVTG,,T,,M,0.078,N,0.144,K,A*33
$GNNGA,220617.00,3933.87070,N,10506.29724,W,1,12,0.95,1735.2,M,-21.7,M,,*4E
$GNGSA,A,3,03,02,24,17,28,06,12,19,,,,,1.63,0.95,1.32*12
$GNGSA,A,3,82,80,81,73,,,,,,1.63,0.95,
```