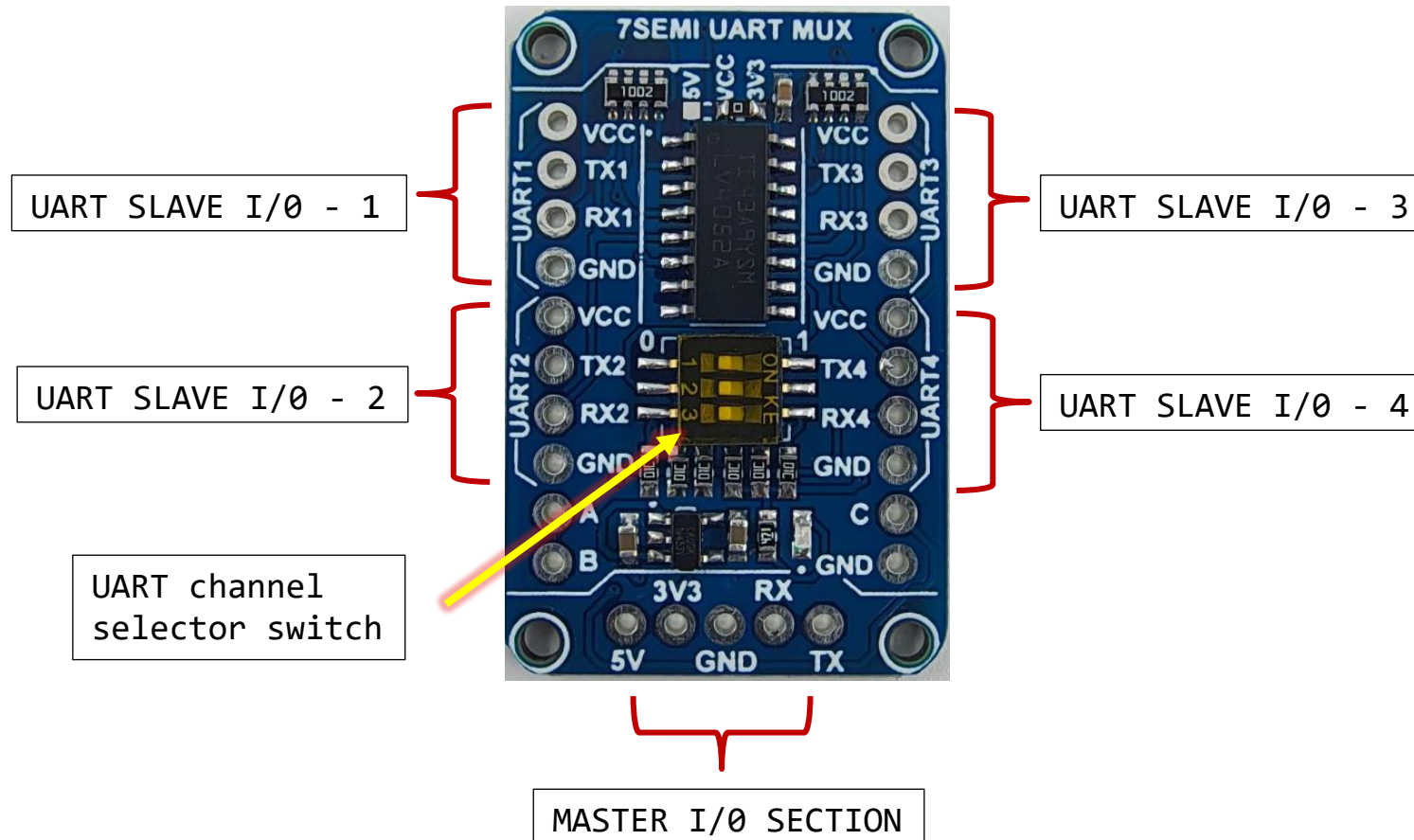


7Semi UART MUX DEMUX USER GUIDE

What this board can do?

- Using this board, you can perform UART communication between one master and multiple slave devices (up to four channels specifically for this board).
- You can do this by changing the switch positions provided on the board.

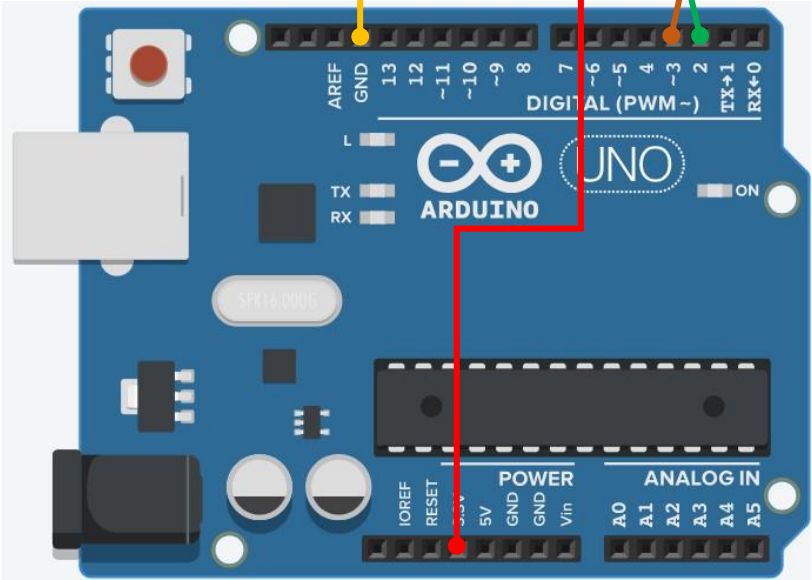
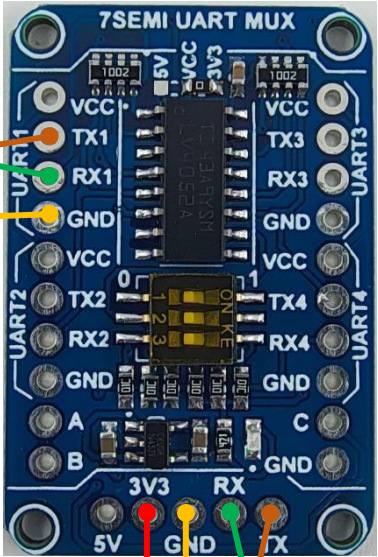
PINOUTS



CONNECTION DIAGRAM OF UART MUX

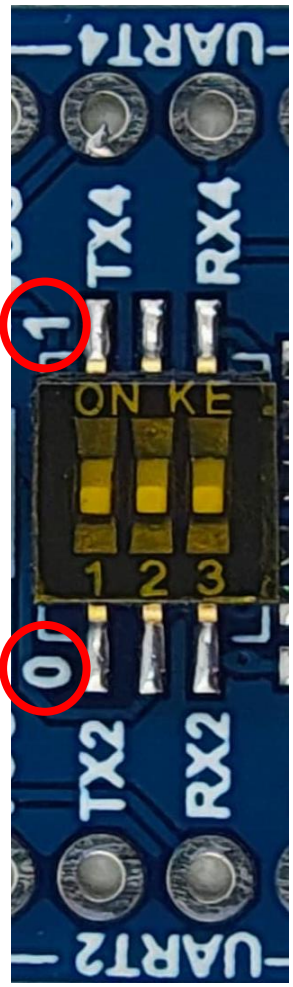


USB to TTL converter



Logic of 'A'	Logic of 'B'	UART output selected
0	0	UART-1
0	1	UART-3
1	0	UART-2
1	1	UART-4

Note:- Logic of C (INH) should be kept low in order to switch on the module



By default, all the switches are connected to Logic Low '0'.

- Switch 1 is connected to A
- Switch 2 is connected to B
- Switch 3 is connected to C

Make the connections as shown in the diagram above and upload the code in your Arduino board given in the link - [UART mux test code](#)

EXPLANATION OF THE ARDUINO CODE TESTING

1. For this particular code of testing we are considering Arduino board as 'MASTER' and USB to TTL converter as 'SLAVE'.
2. The UART communication will now take place between Arduino and USB to TTL.
3. For this testing we need to open two serial monitors in the Arduino IDE.
4. First serial monitor will open at the COM port to which your Arduino board is connected.
5. To open the second serial monitor in your Arduino IDE go to **File → New sketch**
6. After connecting USB to TTL converter to your Computer/ Laptop USB you will get one more COM port. To check the port you can open **Device Manager** in your system and search under the option **PORTS (USB & LPT)**.
7. In the second serial monitor select the COM port of the USB to TTL for which you searched in the device manager.
8. The serial monitor at USB to TTL COM port should display '**UART, Hello 1 to UART, Hello 5**' after pressing the reset button of the Arduino board.
9. Now to send message from second serial monitor to first one you can type message in the second serial monitor and press enter.
10. You can also follow the images provided below for reference.

STEP-1:- Open both serial monitors

The image shows two side-by-side windows of the Arduino IDE 2.3.2. The left window is titled 'UART_MUX_TEST_3 | Arduino IDE 2.3.2' and shows a sketch named 'UART_MUX_TEST_3.ino'. The code includes `SoftwareSerial` and sets up a serial port. The right window is titled 'sketch_jul16a | Arduino IDE 2.3.2' and shows a sketch named 'sketch_jul16a.ino' with a basic `setup()` and `loop()` structure. Both windows have their 'Serial Monitor' tabs open. The left monitor is set to 'COM8' and the right to 'COM3'. Both are set to 'Both NL & CR' and '9600 baud'. Red boxes highlight the port names 'COM8' and 'COM3' in the status bars at the bottom of each window. Green text boxes with black borders are overlaid on the Serial Monitor areas, containing the text 'Serial monitor of Arduino COM port' and 'Serial monitor of USB to TTL COM port' respectively.

```
UART_MUX_TEST_3.ino
1 #include<SoftwareSerial.h>
2 SoftwareSerial myUART(2, 3); //Rx, Tx
3
4 int i;
5 void setup() {
6   Serial.begin(9600);
7   myUART.begin(9600);
8   delay(5000);
9   for(i = 1; i < 6; i++)
10  {
```

```
sketch_jul16a.ino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

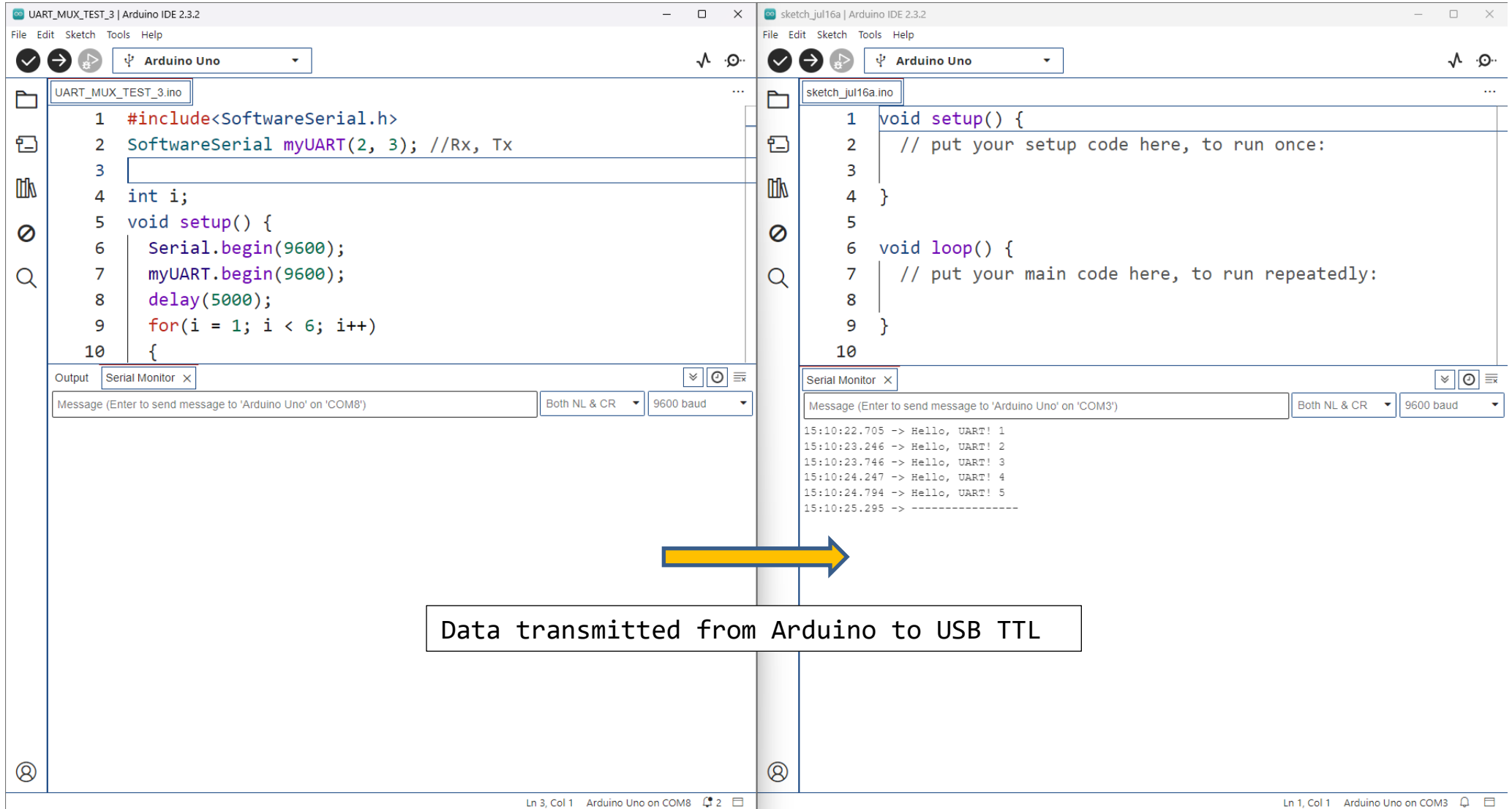
Serial monitor of Arduino COM port

Serial monitor of USB to TTL COM port

Ln 3, Col 1 Arduino Uno on COM8

Ln 1, Col 1 Arduino Uno on COM3

STEP-2:- Press the reset button of your Arduino board. Data will be transmitted from Arduino to USB TTL



The image displays two side-by-side windows of the Arduino IDE 2.3.2. The left window, titled 'UART_MUX_TEST_3 | Arduino IDE 2.3.2', shows the code for 'UART_MUX_TEST_3.ino'. The code includes `SoftwareSerial.h`, initializes a `SoftwareSerial` object named `myUART` on pins 2 and 3, and contains a `setup` function that starts the serial port at 9600 baud, a 5000ms delay, and a `for` loop that prints 'Hello, UART!' five times. The right window, titled 'sketch_jul16a | Arduino IDE 2.3.2', shows the code for 'sketch_jul16a.ino', which is a simple sketch with a `setup` function and a `loop` function. The serial monitor in the right window shows the output of the code, displaying five lines of 'Hello, UART!' with timestamps. A yellow arrow points from the serial monitor of the right window to a text box below it.

```
1 #include<SoftwareSerial.h>
2 SoftwareSerial myUART(2, 3); //Rx, Tx
3
4 int i;
5 void setup() {
6   Serial.begin(9600);
7   myUART.begin(9600);
8   delay(5000);
9   for(i = 1; i < 6; i++)
10  {
```

```
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

15:10:22.705 -> Hello, UART! 1
15:10:23.246 -> Hello, UART! 2
15:10:23.746 -> Hello, UART! 3
15:10:24.247 -> Hello, UART! 4
15:10:24.794 -> Hello, UART! 5
15:10:25.295 -> -----

Data transmitted from Arduino to USB TTL

STEP-3:- Data transmitting from USB TTL to Arduino board. Press enter after typing your message.

The image displays two side-by-side screenshots of the Arduino IDE 2.3.2 interface. The left window, titled 'UART_MUX_TEST_3 | Arduino IDE 2.3.2', shows the code for 'UART_MUX_TEST_3.ino'. The code includes `SoftwareSerial.h`, initializes `myUART(2, 3)`, and contains a `setup()` function that starts the serial port at 9600 baud, delays for 5000ms, and then sends the message 'Hello' followed by 'Electronics' in a loop. The Serial Monitor at the bottom shows the output: '15:19:38.806 -> Hello', '15:19:38.806 ->', '15:19:59.770 -> Electronics', and '15:19:59.770 ->'. The right window, titled 'sketch_jul16a | Arduino IDE 2.3.2', shows the code for 'sketch_jul16a.ino'. It features a `setup()` function and a `loop()` function that sends the message 'Hello, UART!' followed by 'UART! 1' through 'UART! 5' in a loop. The Serial Monitor at the bottom shows the output: '15:10:22.705 -> Hello, UART! 1', '15:10:23.246 -> Hello, UART! 2', '15:10:23.746 -> Hello, UART! 3', '15:10:24.247 -> Hello, UART! 4', '15:10:24.794 -> Hello, UART! 5', and '15:10:25.295 -> -----'. A yellow box highlights the 'Message' input field in the Serial Monitor of the right window, with a text box above it saying 'Type your message here'. A yellow arrow points from this area towards the bottom center, where a text box reads 'Data transmitted from USB TTL to Arduino'.

```
1 #include<SoftwareSerial.h>
2 SoftwareSerial myUART(2, 3); //Rx, Tx
3
4 int i;
5 void setup() {
6   Serial.begin(9600);
7   myUART.begin(9600);
8   delay(5000);
9   for(i = 1; i < 6; i++)
10  {
```

```
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

Message (Enter to send message to 'Arduino Uno' on 'COM8') Both NL & CR 9600 baud

15:19:38.806 -> Hello
15:19:38.806 ->
15:19:59.770 -> Electronics
15:19:59.770 ->

Message (Enter to send message to 'Arduino Uno' on 'COM3') Both NL & CR 9600 baud

15:10:22.705 -> Hello, UART! 1
15:10:23.246 -> Hello, UART! 2
15:10:23.746 -> Hello, UART! 3
15:10:24.247 -> Hello, UART! 4
15:10:24.794 -> Hello, UART! 5
15:10:25.295 -> -----

Type your message here

Data transmitted from USB TTL to Arduino

Ln 3, Col 1 Arduino Uno on COM8

Ln 1, Col 1 Arduino Uno on COM3

