# Using the Programming Arduino

Author: Mark Olson --- GitHub branch KCX\_BT\_EMITTER\_V1.7

<https://github.com/Mark-MDO47/BluetoothAudioTransmitter_KCX_BT_EMITTER/tree/KCX_BT_EMITTER_V1.7>

<https://github.com/Mark-MDO47/BluetoothAudioTransmitter_KCX_BT_EMITTER/blob/KCX_BT_EMITTER_V1.7/ProgrammingArduino/ProgrammingArduino.ino>

We use the Programming Arduino to program the VMLINK table in the KCX\_BT\_EMITTER Bluetooth Audio Transmitter Module. VMLINK is the table that stores the info on Bluetooth receiver(s) (speaker, headphone, etc.) that the KCX\_BT\_EMITTER would automatically connect to. This KCX\_BT\_EMITTER VMLINK table can store info about more than one Bluetooth receiver. If info about more than one Bluetooth receiver is stored in VMLINK, the KCX\_BT\_EMITTER would try to connect to the first entry that was a device that it could see on its scan of Bluetooth devices.

## Connections

Image of laptop USB connected to Arduino with 4 wires connected to KCX_BT_EMITTER; bluetooth speaker on the side.

The Programming Arduino should be a type of Arduino that uses 5 Volt interfaces. For example, an Arduino Uno or an Arduino Nano Classic.

## Programming

* Power off programming Arduino by disconnecting from USB
* Connect wires as follows

|  |  |  |
| --- | --- | --- |
| Arduino Pin | KCX\_BT\_EMITTER pin | suggested wire color |
| 5V | +5V | Red |
| GND | PGND | Black |
| D2 (TX) | RX | Green |
| D9 (RX) | TX | Yellow |

* Connect programming Arduino to USB for PC running the Arduino software
* On the PC running the Arduino software
* Upload the sketch from ProgrammingArduino.ino into the programming Arduino
* Open Serial Monitor by selecting menu "Tools" -> "Serial Monitor"
* Follow instructions on the serial monitor
* After each selected step, wait for the string "--- KCX\_BT\_EMITTER PROGRAMMING STEP COMPLETE ---"
* Disconnect programming Arduino from USB for PC running the Arduino software

## Sample Session

For this sample session, we start with the “Old and Broken” device in the VMLINK table. We want to remove that and put in our “S1 Pro” device. Because both Jim and Mark have S1 Pro Bluetooth speakers, I will label this one “S1 Pro MDO” (you do not need to use the default name provided by the manufacturer).

In order to add S1 Pro MDO we need to know what its unique address is. This can be found by turning the speaker on and telling the KCX\_BT\_EMITTER to scan for Bluetooth speakers and headphones that it can connect to.

The table below shows the Serial Monitor output from a session of programming the KCX\_BT\_EMITTER Bluetooth Audio Transmitter module. The colors for the serial monitor output column are:

* BLACK - communication from the Programming Arduino, either asking for directions or giving feedback. It often asks which “programming step” to execute: SCAN, DISPLAY, ADD, or DELETE ALL.
* RED - “AT” commands sent to the KCX\_BT\_EMMITER. It takes several “AT” commands to perform a user-selected “programming step”.
* GREEN - KCX\_BT\_EMMITER direct status response to the “AT” command.
* BLUE - communication from the KCX\_BT\_EMMITER reporting what it sees on its scan.

| Programming Arduino Serial Monitor output | Comments |
| --- | --- |
| Bluetooth Programming Arduino init... completed!  0 - Pair with Bluetooth receiver devices (such as speaker, headphones, etc.)  1 - Scan for Bluetooth receiver devices  2 - Display stored auto-connect Bluetooth receiver devices  3 - Add one auto-connect Bluetooth receiver device to storage  4 - Delete all auto-connect Bluetooth receiver devices from storage  5 - Current XMTR status  6 - BT Disconnect  7 - PowerOff module  ==> | Startup  Request user to command action  User types in number |
| 5=STATUS | feedback to user on selection |
| SCAN....  SCAN.... | Scan output from KCX\_BT\_EMITTER |
|  |  |
| ------------- RESULT ---------------  --> CMD 0: AT+ | RESULT flag to show start/end  “Aliveness” command |
| OK+ | command response |
|  |  |
| --> CMD 1: AT+GMR? | “query version” cmd to KCX\_BT\_EMITTER |
| OK+VERS:KCX\_BT\_RTX\_V1.4 | command response |
|  |  |
| --> CMD 2: AT+BAUD? | “query baud” cmd to KCX\_BT\_EMITTER |
| OK+BAUD=0,BAUD=9600 | command response |
| SCAN.... | Scan output from KCX\_BT\_EMITTER |
|  |  |
| --> CMD 3: AT+BT\_MODE? | “query mode” cmd to KCX\_BT\_EMITTER |
| OK+BT\_EMITTER | command response |
|  |  |
| --> CMD 4: AT+CHANNEL? | “query chan” cmd to KCX\_BT\_EMITTER |
| OK+CHANNEL=BT CHANNEL | command response |
|  |  |
| --> CMD 5: AT+VOL? | “query vol” cmd to KCX\_BT\_EMITTER |
| OK+VOL=31 | command response |
|  |  |
| --> CMD 6: AT+STATUS? | “query statu” cmd to KCX\_BT\_EMITTER |
| OK+STATUS:0 | command response |
|  |  |
| ------------- RESULT END ----------- | RESULT flag to show start/end |
| SCAN.... | Scan output from KCX\_BT\_EMITTER |
| --- KCX\_BT\_EMITTER PROGRAMMING STEP COMPLETE ---  0 - Pair with Bluetooth receiver devices (such as speaker, headphones, etc.)  1 - Scan for Bluetooth receiver devices  2 - Display stored auto-connect Bluetooth receiver devices  3 - Add one auto-connect Bluetooth receiver device to storage  4 - Delete all auto-connect Bluetooth receiver devices from storage  5 - Current XMTR status  6 - BT Disconnect  7 - PowerOff module  ==> | End of prev commanded steps  Request user to command action  User types in number |
| 1=SCAN | feedback to user on selection |
| SCAN.... | Scan output from KCX\_BT\_EMITTER |
|  |  |
| ------------- RESULT ---------------  --> CMD 0: AT+ | “Aliveness” command |
| OK+ | command response |
|  |  |
|  |  |
| --> CMD 1: AT+RESET | RESET cmd to KCX\_BT\_EMITTER |
| OK+RESET | command response |
| ⸮e  11 | Glitches on serial line |
| POWER ON | command response |
| SCAN.... | Scan output |
|  |  |
| --> CMD 2: AT+ | Aliveness cmd to KCX\_BT\_EMITTER  No longer any SCAN command |
| OK+ | command response |
|  |  |
| --> CMD 3: AT+ | Aliveness cmd to KCX\_BT\_EMITTER  No longer any SCAN command |
| OK+ | command response |
| MacAdd:9dd3ecfd4ef4,Name:S1 Pro | Scan output |
|  |  |
| --> CMD 4: AT+ | Aliveness cmd to KCX\_BT\_EMITTER  No longer any SCAN command |
| OK+ | command response |
|  |  |
| ------------- RESULT END ----------- | RESULT flag to show start/end |
| --- KCX\_BT\_EMITTER PROGRAMMING STEP COMPLETE ---  0 - Pair with Bluetooth receiver devices (such as speaker, headphones, etc.)  1 - Scan for Bluetooth receiver devices  2 - Display stored auto-connect Bluetooth receiver devices  3 - Add one auto-connect Bluetooth receiver device to storage  4 - Delete all auto-connect Bluetooth receiver devices from storage  5 - Current XMTR status  6 - BT Disconnect  7 - PowerOff module  ==> | End of prev commanded steps  Request user to command action  User types in number |
| 2=DISPLAY | feedback to user on selection |
|  |  |
| ------------- RESULT ---------------  --> CMD 0: AT+ |  |
| OK+ |  |
|  |  |
| --> CMD 1: AT+VMLINK? | Query VMLINK cmd |
| OK+VMLINK  BT\_ADD\_NUM=01  BT\_NAME\_NUM=01  Auto\_link\_Add:null  MEM\_MacAdd 00:000000000012  MEM\_Name 00:Old and Broken | Old and Broken device is in the  VMLINK table, but we want to remove  that and put in our S1 Pro device |
|  |  |
| ------------- RESULT END ----------- |  |
| --- KCX\_BT\_EMITTER PROGRAMMING STEP COMPLETE ---  0 - Pair with Bluetooth receiver devices (such as speaker, headphones, etc.)  1 - Scan for Bluetooth receiver devices  2 - Display stored auto-connect Bluetooth receiver devices  3 - Add one auto-connect Bluetooth receiver device to storage  4 - Delete all auto-connect Bluetooth receiver devices from storage  5 - Current XMTR status  6 - BT Disconnect  7 - PowerOff module  ==> |  |
| 4=DELETE ALL |  |
| ------------- RESULT ---------------  --> CMD 0: AT+ |  |
| OK+ |  |
|  |  |
| --> CMD 1: AT+DISCON | RESET command |
| OK+DISCON |  |
| SCAN.... |  |
|  |  |
| --> CMD 2: AT+DELVMLINK | Delete everything in VMLINK |
| Delete\_Vmlink |  |
| MacAdd:9dd3ecfd4ef4,Name:S1 Pro  CON ONE  CONNECT=>MacAdd:9dd3ecfd4ef4,Name:S1 Pro |  |
|  |  |
| --> CMD 3: AT+RESET | RESET so we read and use the new  VMLINK table (all empty now) |
| OK+RESET |  |
| SCAN.... |  |
| e  11 | Glitches on serial line |
| POWER ON |  |
| SCAN....  MacAdd:9dd3ecfd4ef4,Name:S1 Pro  CON LAST |  |
|  |  |
| --> CMD 4: AT+ |  |
| OK+ |  |
|  |  |
| --> CMD 5: AT+VMLINK? | Display VMLINK again |
| OK+VMLINK  BT\_ADD\_NUM=00  BT\_NAME\_NUM=00  Auto\_link\_Add:9dd3ecfd4ef4 | all empty |
| ------------- RESULT END ----------- | RESULT flag to show start/end |
|  |  |
| --- KCX\_BT\_EMITTER PROGRAMMING STEP COMPLETE ---  0 - Pair with Bluetooth receiver devices (such as speaker, headphones, etc.)  1 - Scan for Bluetooth receiver devices  2 - Display stored auto-connect Bluetooth receiver devices  3 - Add one auto-connect Bluetooth receiver device to storage  4 - Delete all auto-connect Bluetooth receiver devices from storage  5 - Current XMTR status  6 - BT Disconnect  7 - PowerOff module  ==> |  |
| 3=ADD | Now we add the S1 Pro MDO to VMLINK |
|  |  |
|  |  |
| Enter the unique MAC address for the Bluetooth speaker or headphones; it starts with 0x  upper or lower case does not matter; maximum of 12 characters after the 0x  To abort adding an auto-connect Bluetooth receiver device to storage, just enter an empty line  ==> |  |
| Your entry "0x9dd3ecfd4ef4" was accepted | Get the address from the SCAN |
|  |  |
|  |  |
| Enter the name you choose for this device; it is OK to place spaces between words  maximum of 20 characters total  To abort adding an auto-connect Bluetooth receiver device to storage, just enter an empty line  ==> |  |
| Your entry "S1 Pro MDO" was accepted | Just about any name you want |
| ------------- RESULT ---------------  --> CMD 0: AT+ | RESULT flag to show start/end  “Aliveness” command  Now we do a command sequence to  add that BT device to VMLINK |
| OK+ |  |
|  |  |
| --> CMD 1: AT+DISCON |  |
| OK+DISCON |  |
|  |  |
| SCAN....  MacAdd:9dd3ecfd4ef4,Name:S1 Pro  CON LAST |  |
|  |  |
| --> CMD 2: AT+VMLINK? | Display VMLINK |
| OK+VMLINK  BT\_ADD\_NUM=00  BT\_NAME\_NUM=00  Auto\_link\_Add:9dd3ecfd4ef4 | There is nothing in VMLINK before  we do our ADD |
|  |  |
| --> CMD 3: AT+ADDLINKADD=9dd3ecfd4ef4 | ADD the MAC Address |
| OK+ADDLINKADD=9dd3ecfd4ef4  BT\_ADD\_NUM=01  BT\_NAME\_NUM=00  Auto\_link\_Add:9dd3ecfd4ef4  MEM\_MacAdd 00:9dd3ecfd4ef4 |  |
|  |  |
| --> CMD 4: AT+ADDLINKNAME=S1 Pro MDO | ADD our name - does not have to  match the name the manufacturer  gave it |
| OK+ADDLINKNAME=S1 Pro MDO  BT\_ADD\_NUM=01  BT\_NAME\_NUM=01  Auto\_link\_Add:9dd3ecfd4ef4  MEM\_MacAdd 00:9dd3ecfd4ef4  MEM\_Name 00:S1 Pro MDO |  |
|  |  |
| --> CMD 5: AT+RESET | We RESET to force it to read and  use the modified VMLINK |
| OK+RESET |  |
| SCAN.... |  |
| ?d  11 | Glitches on serial line |
| POWER ON | command response |
| MacAdd:9dd3ecfd4ef4,Name:S1 Pro  CON MATCH ADD  CONNECT=>MacAdd:9dd3ecfd4ef4,Name:S1 Pro | Note that it matched the MAC address  even though we had a different  name. It still connected. |
|  |  |
| --> CMD 6: AT+ |  |
| OK+ |  |
|  |  |
| --> CMD 7: AT+VMLINK? | Make sure we put the right stuff  in the VMLINK |
| BT\_ADD\_NUM=01  BT\_NAME\_NUM=01  Auto\_link\_Add:9dd3ecfd4ef4  MEM\_MacAdd 00:9dd3ecfd4ef4  MEM\_Name 00:S1 Pro MDO | OK that is from our ADD |
|  |  |
| ------------- RESULT END ----------- |  |
| --- KCX\_BT\_EMITTER PROGRAMMING STEP COMPLETE ---  0 - Pair with Bluetooth receiver devices (such as speaker, headphones, etc.)  1 - Scan for Bluetooth receiver devices  2 - Display stored auto-connect Bluetooth receiver devices  3 - Add one auto-connect Bluetooth receiver device to storage  4 - Delete all auto-connect Bluetooth receiver devices from storage  5 - Current XMTR status  6 - BT Disconnect  7 - PowerOff module  ==> |  |