Using the Programming Arduino

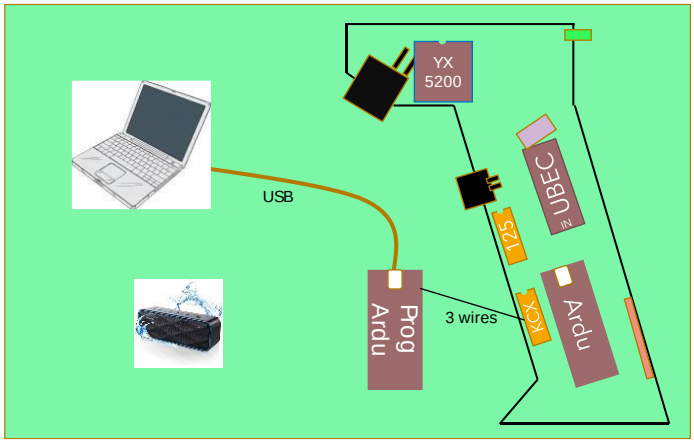
Author: Mark Olson

https://github.com/Mark-MDO47/RubberBandGun

https://github.com/Mark-MDO47/RubberBandGun/blob/master/RBG_arduino/ProgrammingArduino/ProgrammingArduino.ino

We use the Programming Arduino to program the VMLINK table in the KCX_BT_EMITTER Bluetooth Audio Transmitter Module for the Rubber Band Gun (RBG). VMLINK is the table that stores the info on Bluetooth speakers that the KCX_BT_EMITTER would automatically connect to. This KCX_BT_EMITTER VMLINK table can store info about more than one Bluetooth receiver (speaker, headphone, etc.) device. 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



Connect the Programming Arduino

1. On Rubber Band Gun (RBG)
   - Power off RBG
   - Remove the clear cover on the handle of the RBG
   - Inside the handle, find the female jumper connectors for the KCX_BT_EMITTER chip (near front of handle)
     - label color of wire
     - GND BLACK
     - 2 RX GREEN
     - 9 TX YELLOW
2. On separate programming Arduino (using +5V interfaces)
   - Power off programming Arduino by disconnecting from USB
   - Connect jumper wires with male ends as follows
     - Pin color of wire
     - GND BLACK
     - D2 RX GREEN (Arduino TX)

- D9 TX YELLOW (Arduino RX)

3. Connect the programming Arduino jumper wires to the RBG jumper wires using color as the guide.

Disconnect the Programming Arduino

1. Ensure that programming Arduino is disconnected from USB for PC running the Arduino software
2. On Rubber Band Gun (RBG)
   - Power off RBG
3. On separate programming Arduino (using +5V interfaces)
   - Disconnect the programming Arduino jumper wires from the RBG jumper wires
4. On Rubber Band Gun (RBG)
   - Store the jumper connectors safely in the handle and attach the clear cover on the handle of the RBG

Programming

1. Connect the Programming Arduino (see above)
2. On Rubber Band Gun (RBG)
   - Power on RBG
3. On the separate programming Arduino (using +5V interfaces)
   - Connect programming Arduino to USB for PC running the Arduino software
   - Upload the sketch from ProgrammingArduino.ino into the programming Arduino
   - Open Serial Monitor by selecting menu "Tools" -> "Serial Monitor"
4. Follow instructions on the serial monitor
   - After each selected step, wait for the string "--- KCX_BT_EMITTER PROGRAMMING STEP COMPLETE ---"
5. Disconnect programming Arduino from USB for PC running the Arduino software
6. Disconnect the Programming Arduino (see below)

Sample Session

For this 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! 1 - Scan for Bluetooth receiver devices (such as speaker, headphones, etc.) 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 ==>	Startup Request user to command action User types in number
1=SCAN	feedback to user on selection
ALL Devices=0	Scan output from KCS_BT_EMITTER
--CMD 0 AT+	"Aliveness" command
OK+	command response
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro	Scan output
--CMD 1 AT+REST	RESET cmd to KCS_BT_EMITTER
OK+REST	command response
POWER ON	command response
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro	Scan output
--CMD 2 AT+SCAN	SCAN cmd to KCS_BT_EMITTER
OK+SCAN	
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro	Scan output
--- KCX_BT_EMITTER PROGRAMMING STEP COMPLETE --- 1 - Scan for Bluetooth receiver devices (such as speaker, headphones, etc.) 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 ==>	
2=DISPLAY	
--CMD 0 AT+	
OK+	
ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro New Devices:1,MacAdd:0xf44efdecd39d,Name=S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro	

Programming Arduino Serial Monitor output	Comments
--CMD 1 AT+REST	
OK+REST	
POWER ON	
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro	
ALL Devices=1	
MacAddr=0xf44efdecd39d,Name=S1 Pro	
--CMD 2 AT+VMLINK?	Show the VMLINK info cmd
OK+VMLINK BT_ADD_NUM=1 BT_NAME_NUM=1 Last_Add=0x00000000000 VM_MacAdd0=0x00000000012 VM_Name0=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
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro	
--- KCX_BT_EMITTER PROGRAMMING STEP COMPLETE --- 1 - Scan for Bluetooth receiver devices (such as speaker, headphones, etc.) 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 ==>	
4=DELETE ALL	
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1	
--CMD 0 AT+	
OK+	
ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro	
--CMD 1 AT+REST	RESET command
OK+REST POWER ON	
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro	
--CMD 2 AT+DISCON	DISCONNECT in case we were connected
OK+DISCON	
ALL Devices=0	
--CMD 3 AT+DELVMLINK	Delete everything in VMLINK
Delete_Vmlink	

Programming Arduino Serial Monitor output	Comments
--CMD 4 AT+REST	RESET so we read and use the new VMLINK table (all empty now)
OK+REST POWER ON	
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro	
--CMD 5 AT+VMLINK?	Display VMLINK again
OK+VMLINK BT_ADD_NUM=0 BT_NAME_NUM=0 Last_Add=0xf44efdecd39d	all empty
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro CONNECTED ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro	NOTE: it says CONNECTED because there is nothing in the VMLINK and it knows **how to** and **did** connect to the S1 Pro speaker. It did not connect before because it had VMLINK and it did not match with the speaker.
--- KCX_BT_EMITTER PROGRAMMING STEP COMPLETE --- 1 - Scan for Bluetooth receiver devices (such as speaker, headphones, etc.) 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 ==>	
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 "0xf44efdecd39d" 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
--CMD 0 AT+	Now we do a command sequence to add that BT device to VMLINK
OK+	
--CMD 1 AT+DISCON	
OK+DISCON DISCONNECT	

Programming Arduino Serial Monitor output	Comments
--CMD 2 AT+VMLINK?	Display VMLINK
OK+VMLINK BT_ADD_NUM=0 BT_NAME_NUM=0 Last_Add=0xf44efdecd39d	There is nothing in VMLINK before we do our ADD
--CMD 3 AT+ADDLINKADD=0xf44efdecd39d	ADD the MAC Address
OK+ ADDLINKADD VM_MacAdd 1 =0xf44efdecd39d	
--CMD 4 AT+ADDLINKNAME=S1 Pro MDO	ADD our name - does not have to match the name the manufacturer gave it
OK+ADDLINKNAME VM_Name 0 =S1 Pro MDO	
--CMD 5 AT+REST	We RESET to force it to read and use the modified VMLINK
OK+REST POWER ON	
New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro New Devices:1,MacAdd:0xf44efdecd39d,Name:S1 Pro	
--CMD 6 AT+VMLINK?	Make sure we put the right stuff in the VMLINK
OK+VMLINK BT_ADD_NUM=1 BT_NAME_NUM=1 Last_Add=0xf44efdecd39d VM_MacAdd0=0xf44efdecd39d VM_Name0=S1 Pro MDO	OK that is from our ADD
CONNECTED ALL Devices=1 MacAddr=0xf44efdecd39d,Name=S1 Pro	It CONNECTED because (1) it came out of RESET and read VMLINK, (2) it found the device, and (3) it matched the VMLINK address
--- KCX_BT_EMITTER PROGRAMMING STEP COMPLETE --- 1 - Scan for Bluetooth receiver devices (such as speaker, headphones, etc.) 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 ==>	