User’s Manual for the AirMini Transmitter

Blueridge Engineering
http://blueridgeengineering.net

Martin Sant
martan@cstone.net
Darrell Lamm
darrelllamm0@gmail.com

August 4, 2019

Contents

1 Introduction

The AirMini transmitter is based on the 16MHz Arduino Pro Mini and uses all open source software. This is an easy-to-assemble kit and gives you access (in the software) to the DCC signal so that you can manipulate it if you wish. The AirMini board can also be programmed to be a Receiver.

See http://blueridgeengineering.net/index.php/wiki/building-the-promini-air/ for details on hardware assembly of the AirMini kit.

You can use any DCC output (say from your Digitraxx or other DCC controller) to transmit out over the Airwire Frequencies (the channel and output power level is adjustable in the software). This does require a small opto-isolated input-board that receives high-voltage, bi-polar DCC and converts it to unipolar, 5V DCC for input to the AirMini.

We will use some terminology that may be new to you. The National Model Railroad Association (NMRA) set forth a standard for communicating with decoders and other model railroad devices called "DCC", short for Digital Command Control. To control and configure these devices, coded, digital voltage waveforms are sent from DCC-compliant throttles over wires (usually) to these, possibly-multiple, devices, all listening to the commands, sifting out which ones are meant for them. These coded waveforms contain digital messages or "packets" that tell the recipient device what to do by means of specifying the "Address" of the device the command is meant for. Most DCC packets are meant for a specific recipient, although a few kinds of messages are meant for *all* listening recipients.

While DCC throttles are mostly concerned with commanding the speed, direction, and other behavior of locomotives, they can also be used to reconfigure the "decoders" that are busy interpreting the DCC commands. Usually, reconfiguration involves changing lighting effects and other behavior of devices on a locomotive, and the so-called "operations mode" (OPS mode) or "on-the-main" mode is very convenient for doing so. Each DCC throttle manufacturer has a slightly different method for putting the throttle into "OPS mode", but once in this mode, they all send the same kind NRMA-compliant DCC packets to re-configure the recipient decoder, by means of changing the *v*alue of a so-called *C*onfiguration Variable (CV).

This last point is the source of a lot of confusion. A *C*onfiguration Variable (CV) is the *fixed address number with a fixed purpose* where we will deliver a *change in its value held at this address*. We usually refer the *fixed address with a fixed purpose* as "CV#", where # is some number. For instance, CV1 holds the value of a device's "short address" (whose *v*alue can be between 1 and 127). The *v*alue stored at this address can be changed, so we often refer to the *v*alue held at the CV address # as "CV#=*v*alue.

2 AirMini Settings/Configuration

The AirMini has a number of default configuration settings that should make it useful "out of the box," and conventional DCC "CV" "OPS mode" or "on the main" re-programming can change these settings.

See Table 1 for the CV's that can be changed to re-configure the Air Mini. To change these values, the throttle the AirMini is connected to must first select the current address

Table 1: AirMini Settings and Configuration

Feature	CVAddress	Valid CV Values	Default	Comments
RF Channel	CV255	0–16	0	Airwire channels
RF Power[†]	CV254	0–10	6	Experimentation is required
Long addr high byte	CV17	192–231	227	Program CV17 *before* CV18!
Long addr low byte	CV18	0–255	40	Default CV17 & CV18 make Address = 9000
Configuration	CV29	0–255	32	CV29=32 to use long address
Short addr	CV1	1–127	3	CV29=0 to use short address
[†] Resets to default on power-up				

of the AirMini, which is 9000 by default. Then according the instructions for the particular throttle you are using, set the throttle in "OPS" or "on-the-main" programming of Configuration Variables (CV's).

The valid range of values for the CV's are specified in Table 1, so if you attempt to set an invalid value for a CV, the entry will be ignored and will NOT take effect! All changes to CV values made in OPS mode are persistent after power-down except for CV254 (RF power level), which will reset to the original default value of 6 upon power-up.

In general, it should not be necessary to change the AirMini's address. But, if you need to change the address, the following information will help you do so. It's a litte complicated. If you are not changing from a short to a long address or vice versa, no change in CV29 is needed. The fifth bit of CV29 specifies whether a "long" (bit 5=1) or "short" (bit 5=0) address is used. For the AirMini, no other bits of CV29 are relevant, so either set CV29=0 to use the "short" address specified in CV1, or set CV29=32 to use the "long" address specied by CV17 and CV18.

When resetting the "long address" for the AirMini, the user *must* set CV17 *before* setting CV18! Once the user programs the value of CV18, the AirMini's address will change to the new address, whose value is set by the funny formula:

$$\text{Address} = (\text{CV17value} - 192) * 256 + \text{CV18value},$$

so for the default values for CV17 and CV18 in Table 1: (227-192)*256 + 40 = 9000. Changing the address by first setting CV17 and then CV18 means that the AirMini will no longer accept OPS Mode changes at the *old* address, and the throttle's address must be changed to the *new* address before going back into OPS mode for any further configuration changes to the AirMini. Similarly, if you change the value of CV29, then the address for the AirMini may be changed from long-to-short or short-to-long address, so you must accordingly change the address on the throttle to communicate with the Air Mini in OPS mode at the new address.

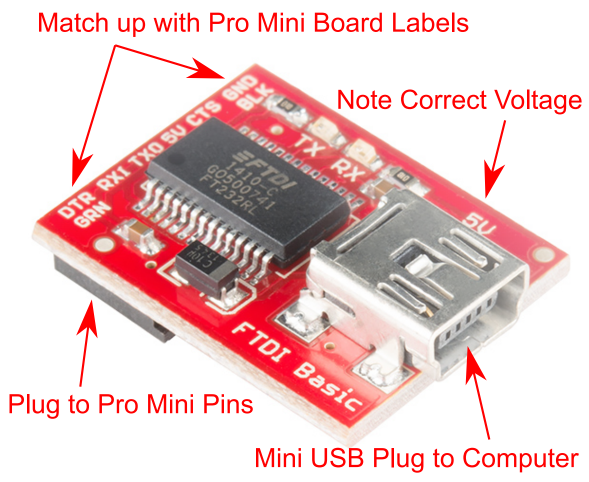
Match up with Pro Mini Board Labels
Note Correct Voltage
Plug to Pro Mini Pins
Mini USB Plug to Computer

Figure 1: 5V Sparkfun FTDI Basic Breakout USB connector

3 Software

The AirMini uses open source software that can be found a `https://github.com/martan3d/AirMini`. Once the source code is down-loaded into a directory that the Arduino development environment can find, it's a snap to compile and down-load updated firmware via USB to the AirMini hardware with the Arduino Integrated Development Environment (IDE).

Since the Pro Mini board in the AirMini does not have it's own USB plug and port to provide a connection to a PC, one end of a *5V* SparkFun FTDI Basic Breakout USB connector can be attached to the six-pin connector on the Pro Mini board and the FTDI's mini USB plug connects to the PC. Be sure to orient the FTDI plug to the Pro Mini pins correctly - match up the labeling on the plug with that of the pins on the Pro Mini board on the AirMini. See Figures 1 and 2.

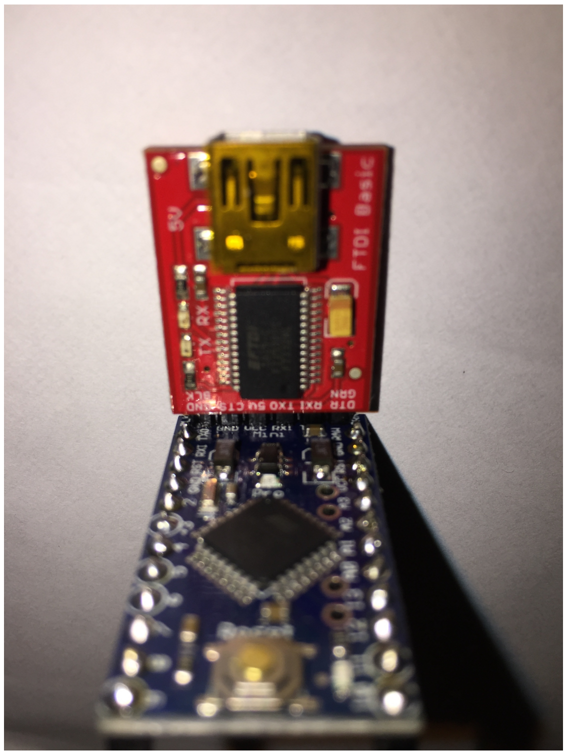

Figure 2: Correct connection of FTDI board to Pro Mini board