ANIMATED HO BUILDING
By Fred Miller, MMR

This article is presented with permission from Model Railroader Magazine.

For almost a dozen years I have been promoting, in various clinics and presentations, the topic of "Light, Sound and Motion" animation for model railroads. Of course over those years more modelers have been venturing into this wonderful aspect of model railroading with a variety of custom implementations. The commercial world now also offers a number of devices and fixtures to bring animation to the layout.

Recently the availability of inexpensive and easy-to-use microcontrollers has opened the animation door even further. The Arduino line of microcontrollers now is a wonderful tool for developing all kinds of custom animation projects.

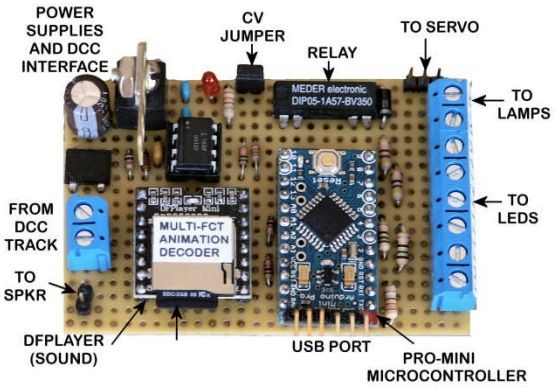
ANIMATED HO SCALE BUILDING

This article describes the installation of one of my animation decoders in an HO Scale building. I have been using the various Arduino microcontrollers for various DCC and Loconet interfaced projects for my model railroad. But I recently decided to put my earlier thoughts of "light, sound and motion" animation into a single custom built DCC decoder. I call my developed project the "Multi-function Animation Decoder" or MFAD. This decoder responds to DCC commands for Functions 0-8 to perform a variety of tasks including control of lights, sounds and servo motions. My intent was to make it easy to animate activities in a variety of buildings on my layout.

In the building described in this article, I have used my MFAD to operate tasks associated with the modeled Import Motors building. The various assigned Function keys:

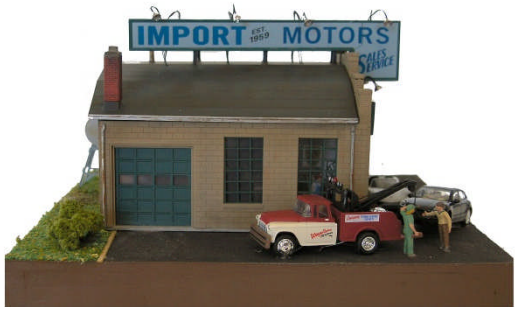
F0 - flicker a LED to simulate a welding operation,
F1 - blink an emergency light on the tow truck,
F2 - turn on lights in the shop area,
F3 - turn on lights in the office,
F4 - turn on outside sign lights,
F5 - play shop sounds
F6 - play welding torch sounds
F7 - play garage door sounds,
F8 - raise or lower garage door

A *YouTube* video of this building in action is available for viewing (See References)



CONSTRUCTING THE BUILDING

The building chosen for this project was the Walther's *Import Motors* (933-4023). The building was built essentially as per instructions except that the back bay and associated garage door was removed, shortening the building. In addition the front windows were rearranged and a garage door was installed instead of one window. The outside plastic lights on the roof sign and front sign were replaced with 10 Miniatronics *N scale Lamp shade with bulbs (*72-001-05). These are 1½ Volt incandescent lamps so they were connected in series to match the



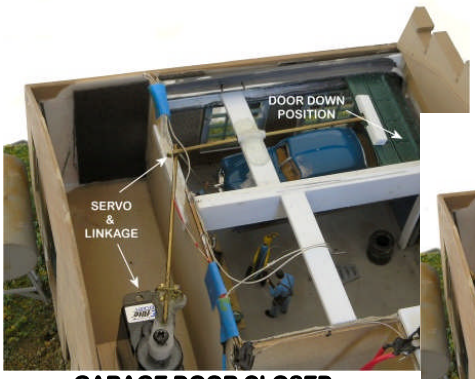
SIDE VIEW SHOWING TRUNCATED STRUCTURE

anticipated 5V provided from the Animation Decoder. Two series of 4 lamps for the roof and one series of 2 plus a 150 ohm resistor worked just fine for the front sign.

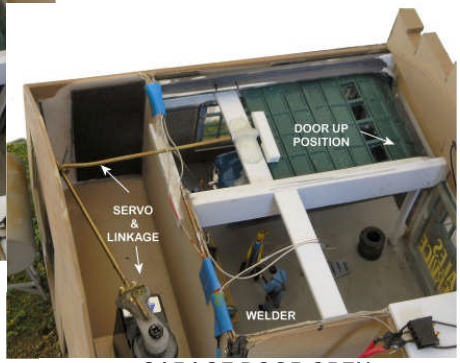
FRONT VIEW WITH REARRANGED DOORS

The inside of the building was partitioned off into an office and shop. Scrap styrene sheet was cut to fit. Styrene was also used to build a car lift and a car was installed in the raised position. Office furniture, shop tools and figures were mounted inside the structure. Small pre-wired SMD LEDs were positioned to light the office, shop and the flickering welding torch up under the raised car.

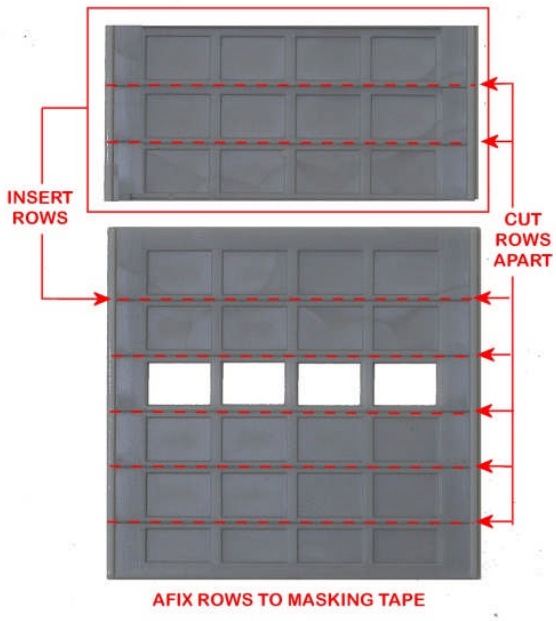
The front garage door was made to open and shut with a micro Servo motor. The doors were constructed from the kit provided doors as shown in the diagram.



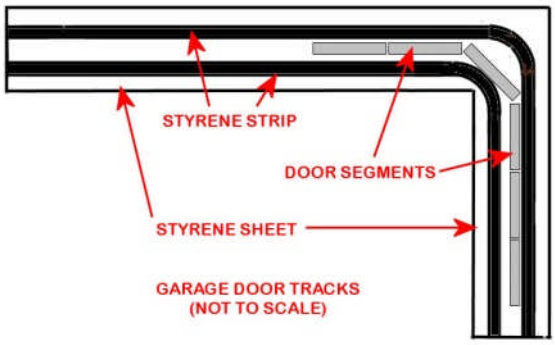
GARAGE DOOR CLOSED



GARAGE DOOR OPEN



AFIX ROWS TO MASKING TAPE



The tracks for the garage door were made from styrene strip and sheet. The strips were heated and bent around a suitable radius to form the tracks with sufficient clearance for the door segments to smoothly pass around corner.

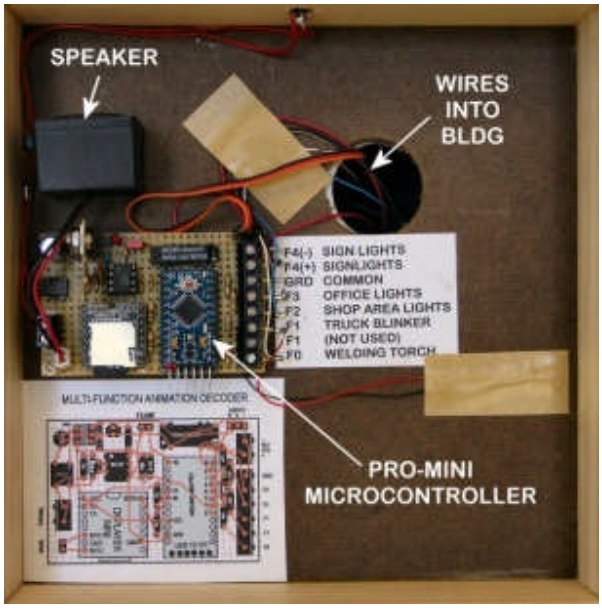
The servo-to-door linkage was formed out of brass rod and positioned for simple movement of the door up and down with the slow moving servo motor.

OPERATING THE ANIMATION DECODER:

The decoders are controlled by standard NMRA DCC decoder function commands. Those commands can be issued from a number of different sources including DCC throttles and JMRI scripts or panels. For my Digitrax LocoNet layout, I have built an Arduino circuit which runs and displays a “day schedule.” A script is defined in the microcontroller which issues LocoNet function ON-OFF commands for specified DCC decoder addresses at particular times of the simulated day. Details about this day scheduler are beyond the scope of this article but available from the author.

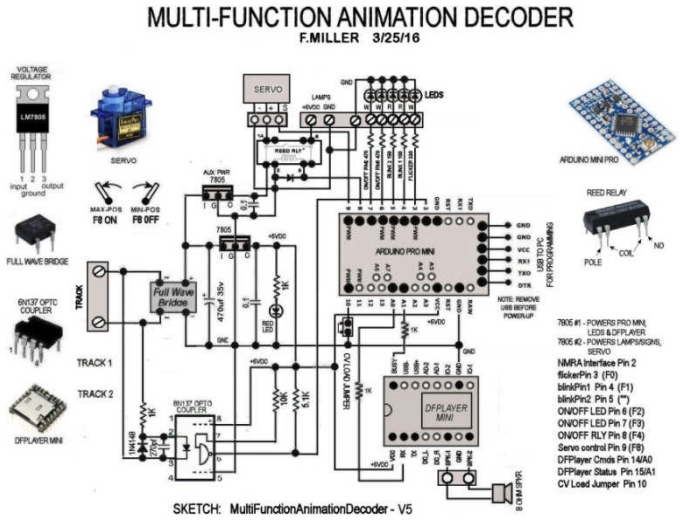
The decoder circuit board is implemented with an Arduino ProMini microcontroller which operates a DFPlayer sound board, playing selected MP3 or WAV format sound tracks, and a micro-servo to provide motion animation. LEDs are driven directly from the ProMini outputs but one additional output drives a relay for more current demanding components, such as the incandescent lamps on the signs.

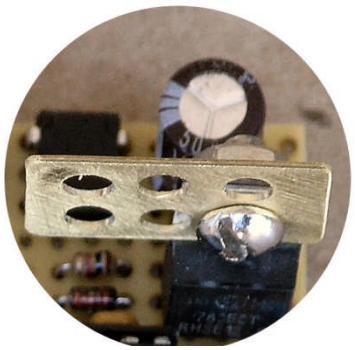
The software running the ProMini microcontroller emulating a DCC decoder can save various option settings as CVs. See the side bar for further details about the CVs in this project.



MFAD BOARD MOUNTED BELOW IMPORT MOTORS BUILDING

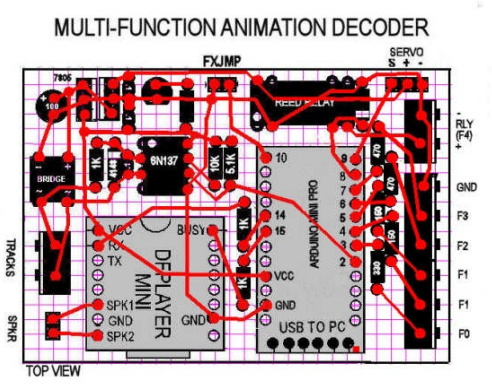
BUILDING THE ANIMATION DECODER

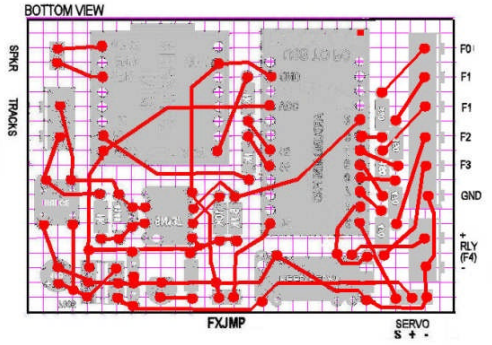




CUSTOM HEAT SINK

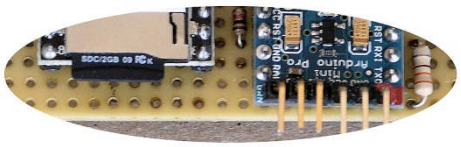
The decoder circuit board accepts power and DCC function commands from the DCC track circuit. I use two 7805 voltage regulator power supply circuits, one for the Arduino ProMini and sound board, the other for powering the Servo and relay operated devices. This provides isolation of possible relay and servo noise from the microcontroller. Since the voltage regulators drop the voltage provided by the DCC track circuit (typically 12 – 15 volts) down to 5VDC they do generate some heat. I use a custom built heat sink to dissipate that heat.





My implementations have used inexpensive parts mounted on a perf board. I use a graphic wiring aid drawn from the basic circuit to mount and wire the parts. Note the wiring aid is first drawn as a top view showing the wiring connections to the parts; then the drawing is flipped to show the actual soldering connections made on the bottom of the board.

The parts are positioned on the board such that the micro SD memory card can be easily inserted/removed from the DFPlayer device to change sound files. Also the programming pins on the Arduino ProMini are positioned to allow access for software modifications.



EASY ACCESS TO MICRO SD CARD & PROGRAMMING PINS

The soft- ware for the

ProMini microcontroller in the MFAD was developed using the standard Arduino development package (IDE) and several provided "libraries" of code which enable simple access and control of devices such as the DFPlayer and the Servo. One essential library was developed by Alex Shepherd for implementation of the NMRA DCC commands and CV controls. Additional work and samples of using the library for DCC decoders was done by Dr. Geoff Bunza. Discussion of these libraries and sample projects is available in the *Arduino for Model Railroads* Yahoo discussion group. I am deeply indebted to these folks for their help and guidance.

A discussion of the software I developed is beyond the scope of this article. However, copies of the programs (called Sketches in the Arduino IDE world) are available from the author.

The cost for each MFAD would be between $10 and $15 depending upon the use of optional components and the sourcing of the parts.

	ANIMATION DECODER PROJECTS PARTS LIST			
QTY	DESCRIPTION	SOURCE	PART #	UNIT COST
2	0.1 UF 25V DISK CAPACITOR	ALLELECTRONICS	104D50	$ 0.12
1	1" SPEAKER IN ENCLOSURE	ALLELECTRONICS	SK-61	$ 2.00
1	100MFD 50V ELECTROLYTIC CAP	ALLELECTRONICS	100R50	$ 0.30
1	270 pf CAPACITOR	ALLELECTRONICS	271D50	$0.06
1	10K OHM 1/4 W RESISTOR	ALLELECTRONICS	291-10K	$ 0.07
2	150 OHM 1/4 W RESISTOR	ALLELECTRONICS	291-150	$0.07
3	1K OHM 1/4 W RESISTOR	ALLELECTRONICS	291-1K	$0.07
1	330 OHM 1/4 W RESISTOR	ALLELECTRONICS	291-330	$ 0.07
2	470 OHM 1/4 W RESISTOR	ALLELECTRONICS	291-470	$ 0.07
1	5.1K OHM 1/4 W RESISTOR	ALLELECTRONICS	291-5.1K	$0.07
1	2GB MICRO SD MEM CARD	ALLELECTRONICS	MSD-2GB	$2.95
2	5-POSITION TERMINAL BLOCK	ALLELECTRONICS	TER-405	$1.40
1	5V REED RELAY	ALLELECTRONICS	RLY-495	$0.60
1	1N4001 DIODE	JAMECO	35975	$ 0.05
1	1N4148 HIGH SPEED DIODE	JAMECO	36038	$0.05
1	2-POS SHORTING JUMPER	JAMECO	112432	$0.15
2	2-POSTION MALE HEADER	JAMECO	103393	$0.09
1	3-POSTION MALE HEADER	JAMECO	103393	$0.14
1	6N137 OPTO ISSOLATOR	JAMECO	113911	$0.75
2	7805 5V REGULATOR	JAMECO	51262	$0.29
1	FULL WAVE BRIDGE	JAMECO	10300	$0.29
1	T1 RED LED	JAMECO	202471	$0.10
1	DFPLAYER MODULE	ALIEXPRESS	-	$2.56
1	MINI-MICRO SERVO SG90	ALIEXPRESS	SG90 SERVO	$1.24
1	PRO-MINI MICROCONTROLLER	ALIEXPRESS		$ 1.32
	PERF BOARD, MISC HARDWARE	-	-	

REFERENCES

YouTube video of animation: https://youtu.be/G675FPWuIGM
Authors website: http://www.fnbcreations.net/projects.htm
Authors email address: tractionfan@aol.com
Arduino Sketches for MFAD projects:
http://www.fnbcreations.net/Articles/MFADSketch.zip
Learning about the Arduino:
https://www.arduino.cc/en/Guide/Introduction
Parts sources: ALLELECTRONICS: http://allelectronics.com
JAMECO: http://jameco.com
ALIEXPRESS: http://aliexpress.com

SIDEBAR – USING THE CONFIGURATION VARIABLES (CVs) FOR THE ANIMATION DECODER

My Animation Decoders operate in a manner similar to DCC throttle controlled locomotive DCC decoders. They have an address recognized by the DCC system in which they are connected and perform the various activities using NMRA DCC function commands. And like the typical throttle decoder, these Animation Decoders can change their operating characteristics by adjusted Configuration Variables (CVs).

The CVs for this project are shown in this table. CVs 30 through 41 specify operating characteristics. As an example CV34 identifies the sound track to play when F5 is activated.

CV	DEFAULT VALUE	CV DEFINITIONS
30	95	SERVO HI POSITION (DOOR UP)
31	78	SERVO LO POSITION (DOOR DOWN)
32	100	BLINK RATE (1/4 ACTUAL SEC)
33	1	PLAY MODE: 0=CONTINUOUS; 1=ONCE
34	1	F5 PLAY TRACK NO.
35	2	F6 PLAY TRACK NO.
36	3	F7 PLAY TRACK NO.
37	10	DEFAULT DCC ADDRESS MSB
38	00	DEFAULT DCC ADDRESS LSB
39	2	PLAY TRACK W/WELDER (0=NONE)
40	1	PLAY TRACK W/DOOR SERVO (0=NONE)
41	25	PLAYBACK VOLUME

The software loaded into each project microcontroller has default values for each of the CVs. If the "CV Jumper" is enabled (plugged in) on the circuit board, the default values will be enabled.

However, if the CV Jumper is removed, the CVs can be changed using the standard NMRA DCC OPS mode CV programming technique. Note that the (long) address is stored in two CVs (MSB and LSB format) separately from a normal decoder location. This makes it possible for the address to be changed (but not activated) in the OPS mode. In the example CVs, the default address is 1000. After new CV values have been programmed by the DCC system, the CV Jumper should be left removed, and the decoder reset to activate the new values. Pressing the MiniPro Reset button (or a power OFF/ON cycle) will reset the microcontroller.

Typical programming steps: (1) with decoder powered, remove CV jumper, (2) do OPS mode programming to "current" address, (3) leaving the jumper out, do a reset with the MiniPro Reset button, (4) leave jumper out for operation unless there is a need to restore the default CV values.

The CV values are stored in nondestructive memory (EEPROM) within the microcontroller so the values will be retained even if the decoder is turned off. They are restored each time the decoder is powered up. If the CV Jumper is restored, the default values will again be activated.