Lightwave RF 433MHz Protocol

Aim

- To document the full 433MHz Lightwave RF protocol.
- To present not just a set of commands that work, but explain the boundaries of what works: where there may be multiple alternative ways to achieve the same thing, and where something must be precise.
- More broadly, my aim from this project is to develop a cheap, low power device which could translate simple LWRF commands into complex sequences of messages. A simple example of this would be the ability to press a single button (whether on a mood switch, a wireless switch or a remote control) when I leave the house which would turn off all of my home entertainment devices, turn off my internal lights after 10 seconds, and turn on my external lights for 2 minutes. There are of course many other possibilities.

Approach

- In approaching the problem, I have drawn from a number of other sources to which I am very grateful and without which I wouldn't have even considered it. The key sources, which fully deserve credit for their work, are:
    - https://github.com/roberttidey/LightwaveRF - the source of the Arduino code I use to both transmit and receive LWRF signals. This translates the 10-character commands (see later) into the transmitted waveforms, and vice-versa.
    - http://www.benjiegillam.com/2013/02/lightwaverf-rf-protocol/ - some of the early work I saw which breaks down the LWRF 10-character codes to understand their meaning. A lot of my work has focused on validating and expanding this.
    - http://lightwaverfcommunity.org.uk/forums/ - an excellent LWRF-focused forum, which includes many threads from a range of contributors which have enabled the collaborative protocol analysis to be done.
- Due to the incomplete and often conflicting nature of the above reference sources, I have independently tested and expanded upon what has previously been published online. The only exceptions are where I don't own certain types of LWRF kit, in which case I've added the published information for completeness but note that it's untested.
- All of my testing has all been done using an Arduino, connected to an iMac running the Arduino serial monitor. The tests are all based on my own LWRF kit - I'm making the assumption that it all works in the same way, but some further sensitivity analysis would be needed to assure this.

- I plan to do additional sensitivity analysis around each test, to understand where something has to be exact and where there is scope for some variation in the messages that are sent or received.
- I've tackled this from both the send and receive angle - I want to be able to receive and correctly understand codes sent from other LWRF controllers, as well as transmit to LWRF devices and be reliably understood every time.
- I will aim to identify best practice for using an Arduino (or similar) as part of an LWRF setup. This includes the choice of Remote ID for the Arduino, when to use individual commands versus moods, and how best to send commands to large numbers of LWRF devices at once. This is work-in-progress and not yet included here.

Basic Protocol

Each LWRF instruction consists of a 10-byte (or more specifically 10 nibbles, represented by 10 hexadecimal (0-F) characters) message. Throughout the remainder of this document I will refer to the messages as the 10-character version for simplicity. These messages consists of the following five parts.

Message Element	Byte Position	Meaning
Value	1-2	New state for the target device
Device	3	The ID (relative to the room and remote) of the target device
Command	4	Defines the type of "value" being set
Remote ID	5-9	A unique ID for each remote
Room	10	The ID (relative to the device and remote) of the target device

The five message parts fall into two types.

- **Message Delivery**: the Remote ID, Room and Device together allow LWRF devices to determine whether any message is intended for that device.
- **Message Content**: the Command and Value together determine how the status of the device will be changed.

Message Delivery

A device will only respond to a specific Remote+Room combination (for Moods and All Off commands) or a Remote+Room+Device combination (for commands targeting individual devices) to which it has been paired.

- A device has no concept of being part of a certain Room or Device outside the context of a specific Remote ID. For example, on one Remote a device may be Device 3 in Room 2, while on a second remote (or even a different button on the same remote) it could be Device 8 in Room 1.

- Similarly for Moods, the group of devices controlled together by one Mood switch does not need to be the same group of devices controlled by a different Mood switch. The concept of being part of that Mood is specific to the Remote ID.
  - However, the different Mood settings on a single Mood switch will always target the same group of devices.
- Using an Arduino / other programmable controller, an almost infinite number of virtual remotes and mood switches may be created to meet whatever purpose is required. However, the limit on the number of controllers to which each device can be paired remains constrained.

Together, understanding this section should be sufficient to ensure any LWRF message is targeted at the appropriate device(s), and also to understand where any message came from.

Message Content

If a message is correctly targeted at a device, that device will use the Command and Value parameters to determine how to act on the received message.

There are only three core Command values, listed in the table below.

Command	Byte Value	Meaning
Off	0	Any "off" command
On	1	Any "on" command, including changing dim levels
Mood	2	Any "mood" setting

Additional Command values are used to operate relays and locking functions - these are not yet documented here.

Each Command value is paired with a Value setting, which provides the specific detail of how the device should respond to the message. There are a larger number of possible Values for different device types. These are summarised in the following table.

Function	Cmd	Value		Meaning
Device Off	0	40	64	Turn off an individual device / dimmer
Device Off	0	00	0	Turn off an individual device / dimmer (for dimmers there is a ~1sec delay, used as pre-cursor to dimming down - included for completeness only)
All Off	0	C0	192	Turn off all devices in a room
Device On	1	00	0	Turn on an individual device (dimmers turn on at previous brightness)
Turn Dimmer On	1	1F	31	Turn on dimmer at previous brightness (also turns on on/off devices)

Function	Cmd	Value		Meaning
Set Dim Level	1	81 - 9F	129-159	Turn on (if needed) then set dimmer to a specific brightness (low-high) (no effect on on/off switches)
Increase Dim Level	1	BF	191	Turn on (if needed) then increase dim level by x% (assuming 5%)
Decrease Dim Level	0	A0	160	Reduce dim level by x% (assuming 5%) (no effect if light already off)
Set Mood	2	80 - 84	128-132	Recall Mood settings 1-5

For any Mood setting, the Room parameter should be F. [test whether this matters]