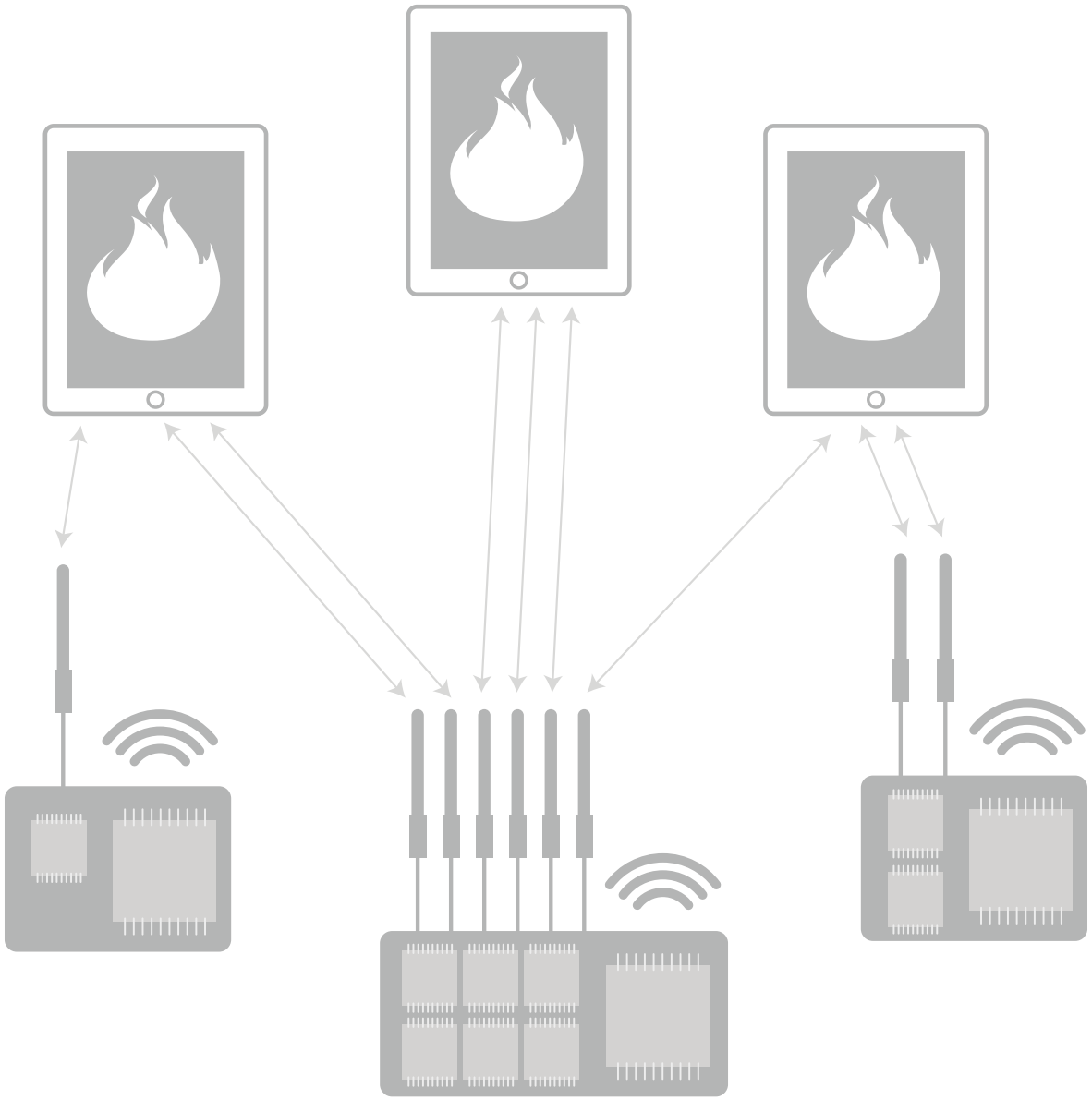
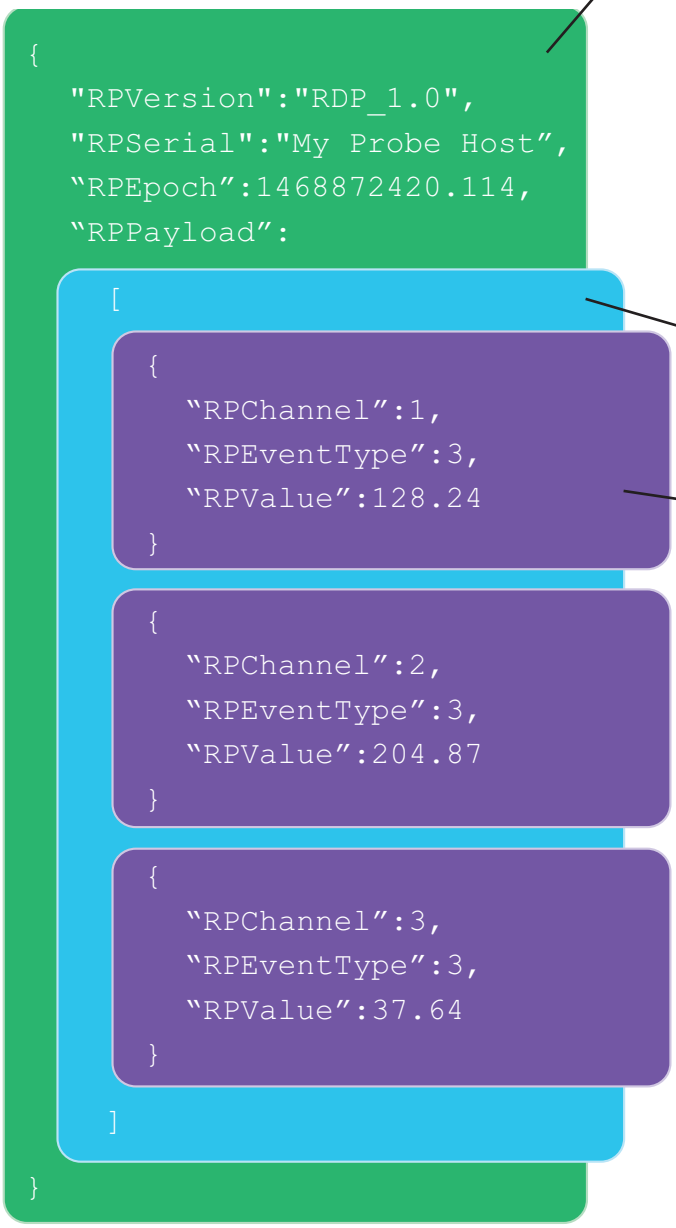
# Roastmaster Datagram Protocol

**RDP DataSheet**



# The Anatomy of an RDP Temperature Transmission

## JSON




**Datagram** *Dictionary*

Key: **RPVersion** Value: **RDP_1.0** *String*

Key: **RPSerial** Value: *String*

*Denotes the serial number (string) with which to associate the payload events. This should match the appropriate probe definition in Roastmaster.*

*Roastmaster will assign an incoming event's data to the probe definition whose serial number and channel match that of the event.*

*You may have one or more hosts, each transmitting information bound for one or more copies of Roastmaster at the same time. The probe definitions in Roastmaster define which events it uses, and which it ignores.*

Key: **RPEpoch** Value: *Integer or Double*

*Counter or time interval type number supplied by host to ensure correct packet order processing in Roastmaster.*

*It is easiest to calculate the number of milliseconds since Unix Epoch Time (Jan 1, 1970) and supply this double-precision float. Alternatively, you can use a simple integer counter, so long as the value is >= 0.*

*If included, Roastmaster will ignore packets that have a lesser value than the last one it processed, ensuring correct ordering.*

*If omitted, Roastmaster will process every packet in the order in which it is received, which is not guaranteed to be the order in which it was sent.*

Key: **RPPayload** Value: *Array of events*

**Payload** *Array of Dictionaries (Events)*

**Event** *Dictionary*

Key: **RPChannel** Value: *Integer*

*Corresponds to the channel defined in the Roastmaster probe definition (1-16).*

Key: **RPEventType** Value: *Integer*
*(Event Type Temperature Constant)*

Key: **RPValue** Value: *Float*

*The probe reading in Celsius. Roastmaster will translate as appropriate, depending on the measurement system set in the curve using the probe.*

**RDP Event Type Integer Constants**

Handshake Sync (SYN) = 1
Handshake Acknowledgment (ACK) = 2,
Temperature = 3,
*//Remaining are not yet published*

# The Anatomy of an RDP Handshaking Transmission

## JSON

```
{
   "RPVersion":"RDP_1.0",
   "RPSerial":"My Probe Host",
   "RPEpoch":1468872420.114,
   "RPPayload":
      [
         {
            "RPEventType":1
         }
      ]
}
```

**RDP packets can be sent as multicast, allowing any instance of Roastmaster on the network to receive its data.**

**It also, though, provides a simple hand-shaking feature to allow the host to discover an instance of Roastmaster configured for its serial name, and send data directly to that device's IP address. This reduces network traffic, and allows users to informally "zone" multiple roasters.**

### Synch (SYN) Datagram

Sent by a host (usually as a multicast) containing one event with the SYN Event Type Constant Integer.

When Roastmaster receives a SYN request, it resets it Epoch value for that serial number, and sends back a corresponding Acknowledgement (ACK) datagram.

**Event** *Dictionary*

Key: **RPEventType** Value: *Integer*
*(Event Type SYN Constant)*

```
{
   "RPVersion":"RDP_1.0",
   "RPSerial":"My Probe Host",
   "RPEpoch":1468872420.114,
   "RPPayload":
      [
         {
            "RPEventType":2
         }
      ]
}
```

### Acknowledgement (ACK) Datagram

Sent by the server (Roastmaster) whenever it receives an SYN datagram. One ACK is sent back to the originating address of each SYN packet it receives.

The host should read this datagram, store the originating address, and switch to a temp sending state.

**Event** *Dictionary*

Key: **RPEventType** Value: *Integer*
*(Event Type ACK Constant)*

**RDP Event Type Integer Constants**

Handshake Sync (SYN) = 1
Handshake Acknowledgment (ACK) = 2,
Temperature = 3,
*//Remaining are not yet published*