pMTC

soo-doh em ti si

An easy to use reader for full frame (SysEx) Midi Timecode

Why Full frame only

- It cuts down on network traffic
- You get all the time information in one packet per frame
- The Kissbox TC2TR supports it out of the box [with minor configuration], which was the original use case.

Change-log

- 0.9.0
    - First logged change
    - Converted to Typescript
    - Enums are now properly named. Old naming will be deprecated in 1.0.X
    - Converted to using Yarn
    - readme updates

Installation

```
yarn install pmtc
```

Usage

```
const { PMTC } = require('pmtc')

const configArgs = {
  // Listen for pMTC data on all interfaces on port 5005
  interfaceAddress: '',
  port: 5005,
  useFreewheel: true,
}

const server = new PMTC(configArgs)
server.run()
server.on('timecode', (data) => {
  console.log(data)
})
```

Want to test with a pMTC Generator? Find one on my Github

Data format

The timecode data is converted to an easy to use JSON packet with a few options.

```
{ "TRANSPORT": "STOPPED", "FRAMERATE": "fr24", "JSON": "
{\"hours\":0,\"minutes\":0,\"seconds\":0,\"frames\":0}", "FRAME": 0, "MTC":
[240, 127, 127, 1, 1, 0, 0, 0, 0, 247], "SEQUENCE": 1560910609673 }
```

Optionally, you can set the mtcOnly flag to receive the raw data packet (useful to multicast or broadcast)

```
const server = new PMTC('', 5005, true)
server.run()

// <Buffer f0 7f 7f 01 01 00 00 03 11 f7>
```

Config Options

interfaceAddress

Description: The IP address of the network interface you want to listen on. **Default:** Any

port

Description: The UDP port to listen on. **Default:** 5005

mtcOnly

Description: Timecode data is sent out exactly as it came in. This is useful for re-broadcasting or adding a freewheel or heartbeat option. **Default:** false

useHeartbeat

Description: Sends out the last known timestamp on an interval if timecode and freewheel aren't running. **Default:** false

useFreewheel

Description: Freewheels internally generated timecode message at the last know frame rate for a predetermined time. **Default:** false

useSequenceNumber

Description: Whether to include a sequence number in the packet. *Note: Sequence numbers will not appear in mtc only packets* **Default:** false

freewheelTolerance

Description: The number of milliseconds past a missed frame should the freewheel kick in. **Default:** 5

freewheelFrames

Description: The number of frames to freewheel before stopping the freewheel. **Default:** 30

heartBeatIntervalMillis

Description: The rate a heartbeat should tick in milliseconds. **Default:** 1000

Functions

PMTC.run()

Starts the server listening for pMTC data.

PMTC.stop()

stops the server.

A note on sequence numbers

But why? Isn't the point of timecode to be sequential?

Yes, however, as this system could potentially be used over UDP, packets do not have a guaranteed delivery or delivery order, this helps ensure that you are not processing old data.

TODOs

- [ ] Add quarter frame support... maybe
- [ ] Fix setters and getters
- [ ] Add a timezone just for fun?