

Bubble と Floaout の仕様書

目次

Bubble (.bub)	1
Bubble のメタデータ	1
Bubble の Block	2
Floaout (.oao)	3
Floaout のメタデータ	3
Bubble のメタデータ	4
Floaout の Block	4
フォーマットを支えるソフトウェア	5
Blower	5
Pop	5
分割スタイル	5
選択スタイル	6
一連の流れの例	6
これらのフォーマットの特徴	7
メリット・デメリット	7
斬新さ	8
終わりに	9
Bubble の表	10
Floaout の表	11

Bubble (.bub)

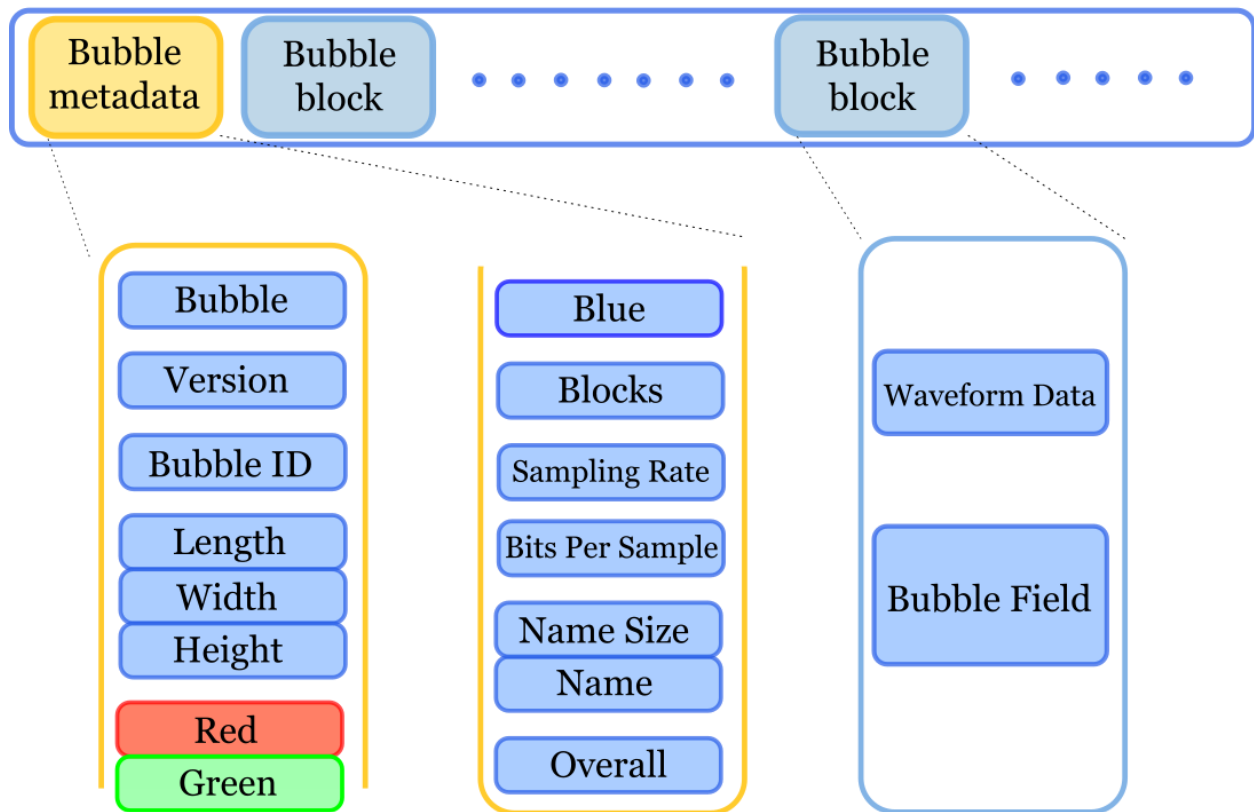


図 1. Bubbleファイルの概要

上記の図のように、Bubble は主にメタデータ、各 Bubble Block の 1 つの楽器や音源のみの波形データと Bubble Field の情報で成り立っています。このフォーマット特有の情報について説明します。

Bubbleのメタデータ

■ Bubble

Bubble は、“bub”という 3 バイトの文字になっており、Bubble フォーマットであることを表しています。

■ Version

Version は、Bubble フォーマットのバージョンを表しています。現在は“0”になっており、このバージョンが変わると互換性がなくなります。

■ Bubble ID

Bubble ID は、リミックスや DJ をする際の著作権問題を解決します。ここに書かれているデータは単なる 16 バイトの数値なのですが、この数値は私が管理するサーバーの情報とリンクしています。そのサーバーには、元のアーティストが指定したリミックスや DJ に対する使用条件の情報があり、それに応じてリミックスや DJ をする方々が使用料等を支払う仕組みになっています。“0”の場合は、リンクされていないことを意味します。

■ Length / Width / Height

これらのデータは、Bubble Field の長さ、幅、高さの情報です。大きさは 2^n で表され、これらのデータは n の部分になります。

■ Red / Green / Blue

これらのデータは、Bubble Field の要素の大きさも加え、RGBA の色にして、音を可視化させるときに使用します。

■ Blocks

Blocks は、含まれている Bubble Block 数を表しています。Bubble Block は、Waveform Data と Bubble Field から成り立ちます。

■ Sampling Rate

一秒間に含まれている波形データの数を表しています。

■ Bits Per Sample

波形データのビット深度を表しています。

■ Name Size

Bubble の名前のバイト数を表しています。

■ Name

Bubble の名前です。

■ Overall

Overall は、Bubble Field の概要です。一目見て Bubble Field のどの部分にどのくらい音が鳴っているかわかるようにします。

BubbleのBlock

■ Waveform Data

Waveform Data は、音の波形データです。この波形データのビット深度は、音の音量等を操作するため、32-bit float または 64-bit float のみに対応しています。

■ Bubble Field

Bubble Field は、指定された Length / Width / Height で 3 次元空間を分割し、分割された要素ごとにおける波形データの音量情報です。3 次元空間の最外殻のみではなく、内側にも要素が存在する、他のフォーマットにはない特有の情報です。

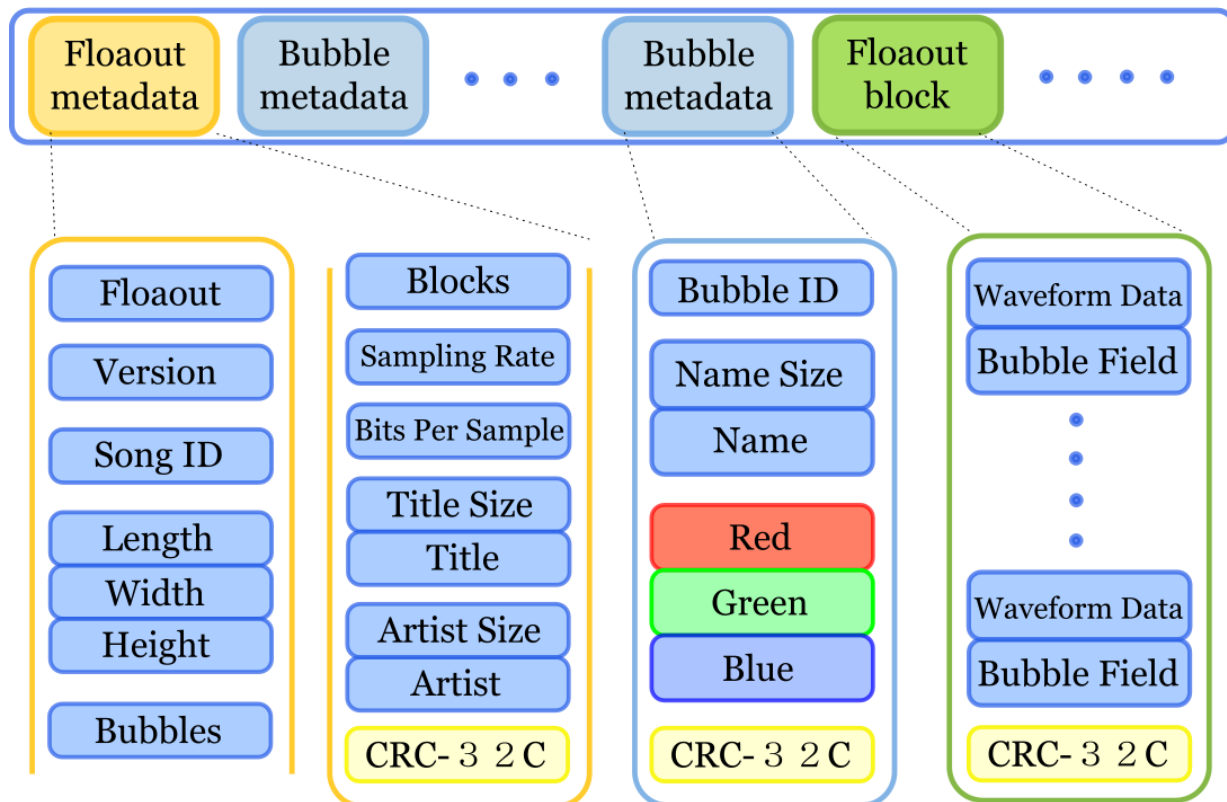
0	255
50	10

0	255
100	50

図 2. Bubble Fieldの例

図 2 の例では、Length が 1、Width が 1、Height が 1 の Bubble Field です。それぞれの要素は 1 バイトの 0~255 の数値になっており、要素の値が 255 である場合は、その位置にあるスピーカーに、波形データを最大音量で鳴らします。

Floaout (.oao)



上記の図のように、Floaout は主にメタデータと含まれている Bubble のメタデータ、各 Floaout Block に含まれる Bubble Block の情報で成り立っています。このフォーマット特有の情報について説明します。

Floaoutのメタデータ

■ Floaout

Floaout は、“oao”という3バイトの文字になっており、Floaout フォーマットであることを表しています。

■ Version

Version は、Floaout フォーマットのバージョンを表しています。現在は“0”になっており、このバージョンが変わると互換性がなくなります。

■ Song ID

Song ID は、私が管理するサーバーの情報とリンクし、作品のタイトルやアーティスト名から歌詞や画像までの様々な曲情報を得ることができます。ファイル自体にそのような情報が含まれていないため、ミュージックビデオや曲の時間と連動した歌詞など、順次情報を追加や変更が可能です。“0”の場合は、リンクされていないことを意味します。

■ Length / Width / Height

これらのデータは、Bubble Field の長さ、幅、高さの情報です。大きさは 2^n で表され、これらのデータは n の部分になります。

■ Bubbles

Bubbles は、Floaout に含まれる Bubble 数を表しています。

■ Blocks

Blocks は、含まれている Floaout Block 数を表しています。Floaout Block は、複数の Bubble Block と CRC-32C から成り立ちます。

■ Title Size

Floaout のタイトルのバイト数を表しています。

■ Title

Floaout のタイトルです。

■ Artist Size

Floaout のアーティスト名のバイト数を表しています。

■ Artist

Floaout のアーティスト名です。

■ Title / Artist

これらのデータは、作品のタイトルとアーティスト名です。

■ CRC-32C

CRC-32C は、誤り検出符号の一種です。データの範囲は、以前の CRC-32C から直前までです。（はじめを除く）

Bubbleのメタデータ

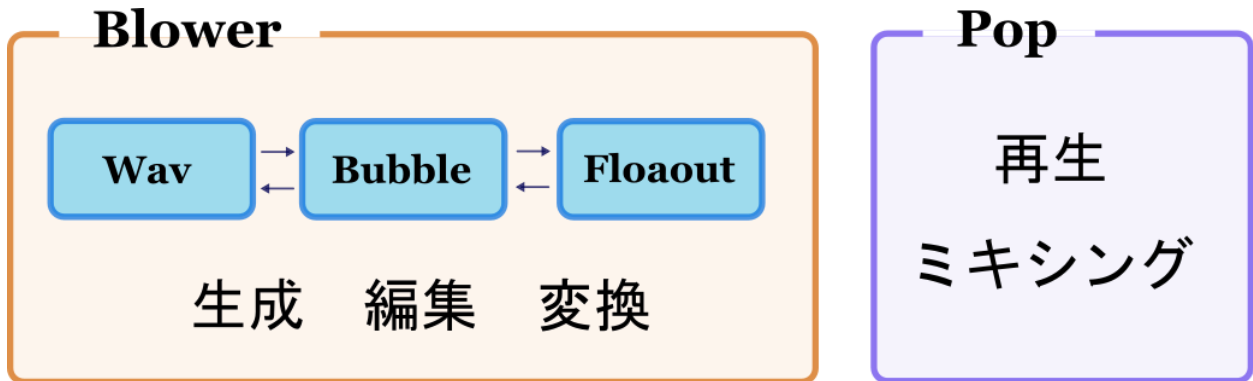
Floaout に含まれる Bubble のそれぞれのメタデータは、Bubble ID、Name Size、Name、Red / Green / Blue、CRC-32C となっています。

BubbleのBlock

Floaout の Block は、Floaout に含まれる Bubble の Block、CRC-32C となっています。

フォーマットを支えるソフトウェア

生成・編集・変換を主とする Blower と再生・ミキシングを主とする Pop の2つのソフトウェアが中心となり、各フォーマットを支えています。



Blower

Blower は生成・編集・変換するソフトウェアです。Blower では、縦軸が Bubble、横軸が時間の編集画面と Floaout に変換する機能を開発します。機能を最小限にとどめることで、多くの方々にプラグインの開発をしてもらうことを期待しています。プラグインの例としては、Wav→Bubble や Floaout→Bubble、Bubble のエフェクト、音量調整などを想定しています。

Pop

Pop は再生・ミキシングするソフトウェアです。Pop では、2種類の再生方法とライブ等で使用するミキシングの機能を開発します。

2種類の再生方法

・分割スタイル

Bubble Field の要素のほうがスピーカーの数より大きい場合、スピーカーの数に合わせて、補間の仕組みを応用して、新しい要素で対応します。

おすすめの使用例). 各家庭での音楽鑑賞

家庭のように、手軽に再生させたい場面で使用してもらうことを想定しています。

メリット

- ・均等に Bubble Field の要素を合成するため、スピーカーの音量調整を意識する必要がない
- ・ユーザーで設定する面倒な調整が不要

デメリット

- ・選択スタイルより自由度が低い

・選択スタイル

それぞれのスピーカーに、再生させたい Bubble Field と Bubble を選択して、その部分のみ再生させます。

おすすめの使用例). ライブ会場での音響環境、映画やテレビの鑑賞

ライブ会場のように、音響環境が特殊であり、細かな設定が必要な場面で使用してもらうことを想定しています。

また、映画やテレビのように、出演している方が複数人いるような場合、人ごとに再生するスピーカーを変えたりすることができます。

メリット

- ・細部まで調整することができて、自由度が高い

デメリット

- ・ Bubble Field や Bubble を選択するまで再生できない

一連の流れの例（ボーカル・ギター・ベース・キーボード・ドラムの5人組ロックバンドの場合）

1. モノラルの音源を複数用意

CD 音源のように曲全体で一つのステレオ wav ファイルを用意するのではなく、音源または楽器ごとにモノラルの wav ファイルを準備します。具体的には、ボーカルの声のみを録音し、その音源をビット深度 32-bit float、サンプリング周波数 96,000Hz のモノラルの wav ファイルにエクスポートします。他のパートも適宜エフェクト等を加え、同様に wav ファイルにエクスポートします。

2. それぞれ Bubble フォーマットに変換

準備した wav ファイルを、Blower で名前や色、Bubble Field の情報を与えて、Bubble ファイルに変換します。

3. つなぎ合わせて Floaout フォーマットに変換

先程変換した Bubble ファイルを、Blower で、まず適宜エフェクトや再生する時間帯の情報を与え、Floaout ファイルに変換します。

4. ライブで再生する

この Floaout ファイルは Pop のミキシング機能によって、5 人では演奏しきれないちょっとしたパートの部分のみを代用することができます。

これらのフォーマットの特徴

Floaout は、チャンネルベースとオブジェクトベースのそれぞれの利点を活かしたイマーシブサウンド対応の今までにない音声フォーマットです。Floaout のメリット・デメリット、他の音声フォーマットとの違いや斬新さをここで紹介します。

メリット

- ・イマーシブサウンド

- Bubble Field の仕組みにより音源に 3 次元の情報が与えられるため、イマーシブサウンドで再生することが可能です。

- ・全チャンネル数に対応

- 2 種類の再生方法により、すべてのスピーカー配置に対応します。

- ・個々の楽器ごとにリアルタイムでミキシング等が可能

- 音源ごとに波形データがあるため、音源ごとにエフェクトやミキシングをリアルタイムですることができます。

- ・高音質

- 波形データのビット深度は 32-bit float または 64-bit float であり、CD で使用されている 16-bit のような他のビット深度とは違い、浮動小数点の計算をするため、音を変化させる際にノイズや誤差が発生しにくく、音質の劣化を最小限に抑えられます。

- ・高品質

- CRC-32C が標準で実装されています。転送等でデータが実は破損しており、オリジナルの音源とは違うのではないかと不安に思う方々にも安心してもらえます。

- Wav のような他のフォーマットは、メタデータにオプションがついているため、読み取りが煩雑になるが、Floaout は統一されているため、開発がしやすくなっています。

デメリット

- ・データサイズが大きい (CD 音源の 2 倍以上)

- 波形データだけでなく Bubble Field 等の他のデータも含まれており、データサイズは大きくなります。しかし情報量が大きいということは、アーティストが伝えたいありのままを伝えられ、いつの時代にも通用する素晴らしい作品ができると考えています。

- ・ファイルの生成方法等が確立していない

- 新フォーマットであるため、生成方法から再生方法までの方法を独自に編み出す必要があります。これからソフトウェア開発を通して、効率の良い方法を探っていきたいです。

チャンネルベースやオブジェクトベースのフォーマットとの違い

Wav や FLAC など、現在のほとんどの音声ファイルはチャンネルベースのフォーマットです。特徴は、予め定めたスピーカーの配置を想定して作られているところです。具体的には、2ch や 7.1ch のようにスピーカーの数がファイルを作成した時点で決まっているため、負荷なく、作成者が望んだありのままの音源を聴くことができます。結果的に、質の高いミキシングがされ、作品の質も向上します。しかし、他の ch では再生することができず、柔軟性に欠けてしまいます。

一方で Dolby Atmos や DTS:X など（完全なオブジェクトベースではない）のイマーシブサウンドの音声ファイルはオブジェクトベースです。特徴は、それぞれの音源に位置情報を与え、それをオブジェクトとしてとらえて、リアルタイムで計算して再生させることです。各映画館のように場所によってスピーカーの数が違う場合でも、リアルタイムでオブジェクトの位置情報に基づいて計算しているため、再生することができます。結果的に、スピーカーの配置にとらわれない柔軟な再生ができます。しかし、作成者の想定していないパンニングで再生される欠点もあります。

これらの利点を活かしたのが Floaout です。Bubble Field によって、作品としての質の高さやリアルタイムでミキシング等をする柔軟性を実現しています。Bubble Field は、各音源の位置情報を与えるのではなく、3次元空間を分割し、それぞれの要素に音源の音量情報を与えます。そのため要素ごとに細かく指定でき、作品の質や自由度が高くなります。

また現在の音響環境は主に、聴く人を中心に円周上や立体の面の部分にスピーカーを配置します。しかし、Bubble Field は3次元全体に音量情報が与えられているため、どのような位置にもスピーカーを配置することができ、音の奥行きを出すことができます。

Song ID

Song ID と似たような機能は、音声再生するソフトウェアに実装されたりしています。しかし、別のソフトウェアでは対応していないので、曲の情報を得ることができません。そのため、フォーマットの段階で取得してもらうことにより、曲の情報を確実に、統一して得ることができます。

Bubble ID

現在は音源を使用する際に、著作権協会に申請し、許可してもらう必要があります。Bubble ID はフォーマットに埋め込まれているため、再生の段階で発見することができ、使用料をいただけます。そのため、使用者は面倒な許諾申請の必要がなく、アーティストは無断で使用されることが少なくなります。

フォーマットを2段階に分ける理由

私は、曲だけでなく各音源にも作品と同等の価値があると思っています。実際にリミックスやDJは、いいメロディーやコードなど、各部分の音源を抽出して新しい曲を作り上げています。しかし、大体的場合は完成した曲の音源から抽出するため、上手く抽出できなかつたりします。また、再生させるときにも、ミキシングが部分ごとにできなかつたりします。このように再利用性を考慮する場合、各部分の音源にもフォーマットをもたせることで、拡張性が生まれると考えました。この考えを基に作り始めたのが Bubble です。

終わりに

直接にはこれらのフォーマットに関係はないのですが、このような思いで、日々開発していることをここに書き記します。

私は音楽が好きではありませんでした。

私が幼稚園のころ、童謡を歌うことがよくありました。しかし、童謡がとても嫌いで口パクをするほどでした。また家族が音楽を聴かなかったこともあり、クラシックぐらいしか他に聴いたことがなく、音楽は音楽理論のようなルールがしっかりしておりつまらないものだと思い、音楽と全く関わっていませんでした。

しかし小学6年生の修学旅行の時、転機が訪れました。移動するためのバス内では、BGMを流していました。その中の一つにサカナクションの「バッハの旋律を夜に聴いたせいです。」という曲があり、その曲はロックとテクノが融合した今までに聞いたことのないジャンルの曲でした。その時、音楽は自由に作ってもいいこととプロがこのような曲を出していたことにとても驚きました。その後無性に聴きたくなり、その曲を聴き始め、サカナクションや他のアーティストの曲も聴くようになりました。そのため私は音楽が嫌いなのではなく、ある特定のジャンルが好きではないだけであったことを気づかされ、音楽への向き合い方も変化していきました。

音楽には直接的には社会に貢献できないかもしれませんが、しかし、音楽はリラクゼーション効果などがあり、私も含め多くの人にヒーリングをしており、生きる上で人生を豊かにしてくれるものだと思います。そのため、今回作成するフォーマットのように音楽の可能性を伝えて、音楽業界のさらなる発展だけでなく、昔の私のように音楽が嫌いだと思っている人たちにも、少しでも音楽に興味をもってくれるように頑張っていきたいです。

表 1. Bubble(.bub)

名称	型 (Bytes)	内容
Bubble	str (3)	“bub” means Bubble
Version	u8 (1)	Bubbleのバージョン (0~255)
Bubble ID	u128 (16)	ファイルのBubble ID
Length	u8 (1)	Bubble fieldの長さ (2^n)
Width	u8 (1)	Bubble fieldの幅 (2^n)
Height	u8 (1)	Bubble fieldの高さ (2^n)
Red	u8 (1)	赤
Green	u8 (1)	緑
Blue	u8 (1)	青
Blocks	u64 (8)	Block数
Sampling Rate	u32 (4)	サンプリング周波数
Bits Per Sample	u16 (2)	ビット深度
Name Size	u8 (1)	Bubbleの名前のサイズ
Name	String	Bubbleの名前 (UTF-8)
Overall	Vec<Vec<Vec<u8>>>	Bubble Fieldの概要

各Bubble Blockにつき

名称	型 (Bytes)	内容
Waveform Data	Sample (4 or 8)	音声データ
Bubble Field	(L × W × H)	Bubble Field

表 2. Floaout(.oao)

名称	型 (Bytes)	内容
Floaout	str (3)	“oao” means Bubble
Version	u8 (1)	Bubbleのバージョン (0~255)
Song ID	u128 (16)	ファイルのSong ID
Length	u8 (1)	Bubble fieldの長さ (2^n)
Width	u8 (1)	Bubble fieldの幅 (2^n)
Height	u8 (1)	Bubble fieldの高さ (2^n)
Bubbles	u16 (2)	Bubble数
Blocks	u64 (8)	Block数
Sampling Rate	u32 (4)	サンプリング周波数
Bits Per Sample	u16 (2)	ビット深度
Title Size	u8 (1)	タイトルのサイズ
Title	String	タイトル (UTF-8)
Artist Size	u8 (1)	アーティストのサイズ
Artist	String	アーティスト (UTF-8)
CRC-32C	u32 (4)	初めからビット深度まで

各Bubbleにつき

名称	型 (Bytes)	内容
Bubble ID	u128 (16)	BubbleのBubble ID
Name Size	u8 (1)	Bubbleの名前のサイズ
Name	String	Bubbleの名前 (UTF-8)
Red	u8 (1)	赤
Green	u8 (1)	緑
Blue	u8 (1)	青
CRC-32C	u32 (4)	一つ前のCRC-32Cから直前まで

各Floaout Blockにつき

名称	型 (Bytes)	内容
Waveform Data	Sample (4 or 8)	1番目のBubbleの音声データ
Bubble Field	(L × W × H)	1番目のBubbleのBubble Field
:	:	:
Waveform Data	Sample (4 or 8)	n番目のBubbleの音声データ
Bubble Field	(L × W × H)	n番目のBubbleのBubble Field
CRC-32C	u32 (4)	一つ前のCRC-32Cから直前まで

n = Bubble 数