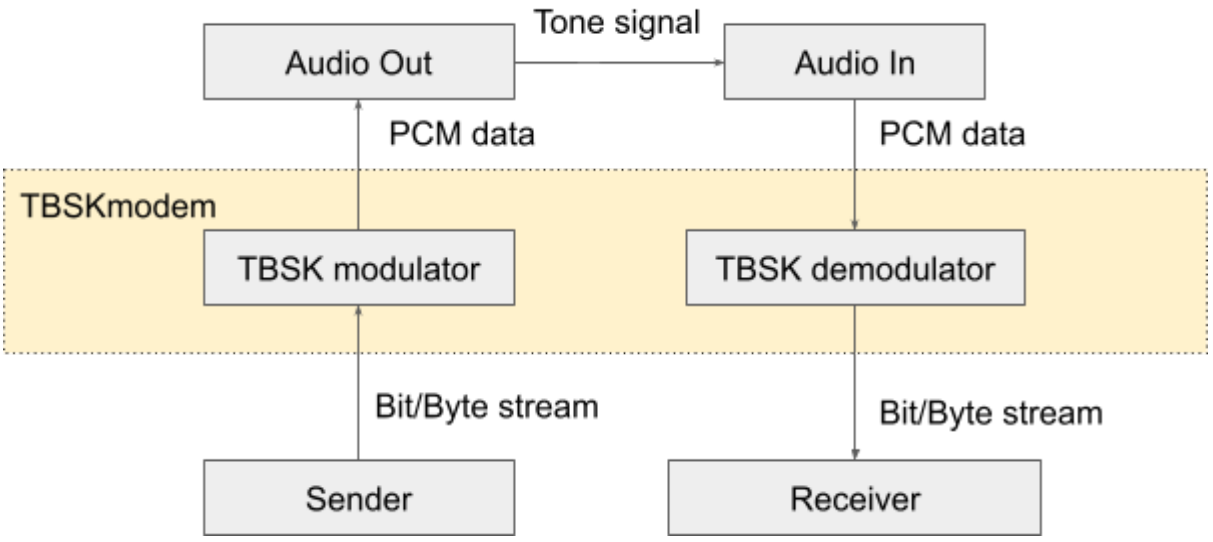
TBSKmodem Rev4

2022-2023 nyatla.jp
https://nyatla.jp/

TBSKmodemは、空気中の短距離音波通信システムである。専用の通信機器を必要とせず、スマートフォン、パソコンのオーディオデバイスの他、マイコンのアナログI/Oで動作する。実用的な最大伝送速度は1k bpsである。エラー訂正機能、検出機能は持たない。



この文章では、TBSKmodemの変調方式、信号型式、複変調手順、実装について説明する。

概略 2
信号型式 4
搬送波型式 4
変調方式 4
TBSKフレーム 5
プリアンブル部 5
ペイロード部 6
ウォームアップ/クールダウン 7
変調復調手順 8
変調 8
トーン信号の生成 8
TBSKフレームの作成 9
シンボル列とトーン信号から変調信号を作成 10
ウォームアップ、クールダウン信号の作成 11
完成したTBSKmodem信号 11
復調 12
相関値分析 12
Positiveサブフィールドの検出 13
Negativeサブフィールドの検出 13
同期位置の確定 13
プリアンブルの評価 14
ペイロードの取得 14
終端の推定 15
性能評価 16
通信速度 16
通信距離 16
環境特性 16
ノイズ特性 16
エラー特性 16
実装 17
コア実装 17
Python 17
Java 17
C# 17
C++ 18
Micro 18
ラップライブラリ 18
Processing 18
Javascript 18
ライブデモ 18

概略

TBSKmodemは、TBSK変調方式により変調した音波により低ビットレートな近距離デジタル伝送を行う。伝送性能は、最大1k Baud で1bit/symbolを伝送する。代表的なビットレートは80, 160, 320, 480, 960bpsである。

他の方式と比較して周波数当たりの情報量は少ないが、変復調に周波数変換処理が不要であり、ビット列の展開演算、数bit精度の相関値演算で実装することができる。復調では、振幅信号を遅延検波して、相関値からシンボルを直接復元する。

搬送波は、任意の周波数帯域幅をもつ特徴のあるトーン信号である。これを差分位相変調して情報を伝達する。周波数の減衰特性が不明、又は不安定な媒体、送受信機でも伝送ができるように設計されている。

トーン信号は、任意形状の振幅信号である。形状は単一周波数のSin波、矩形波、合成波等を使用できる。通常は、これらの基本信号を広帯域に拡散して使用するが、そのまま使うこともできる。

振幅信号は、通信路に使われる空間と周波数帯を占有する。時分割以外の多重化は想定していない。

TBSKmodemでは、エラー検出、訂正なしのコネクションレス型の回線である。これらの課題、衝突制御、コネクション等については、上位レイヤで解決する。

信号型式

TBSKmodemの搬送波型式、シンボルの型式、変調、復調方式について説明する。

搬送波型式

搬送波は、特徴のあるトーン信号をT secの期間で区切った単位トーン信号で構成する。特徴のあるトーン信号とは、一定値でない信号である。

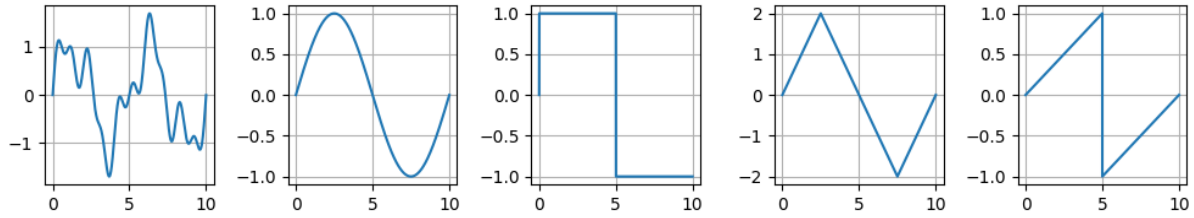


図. 単位トーン信号の例

単位トーン信号は連続しており、TBSK搬送波を形成する。単位トーン信号の形状は任意であるが、1つのTBSK搬送波では、全体で同じ単位トーン信号を使用しなければならない。

変調方式

TBSKmodemでは、単位トーン信号と、単位トーン信号の位相を反転した2種のシンボルを定義する。元の信号をシンボルP、反転した信号をシンボルNとする。シンボルの長さは、単位トーン信号と同一である。

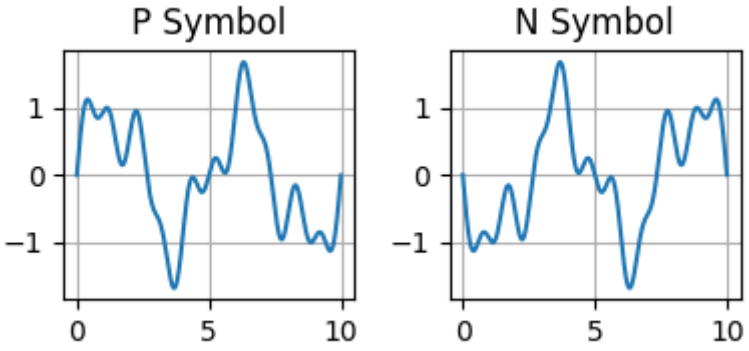


図.PシンボルとNシンボルの例

シンボル値[n]は差動値である。隣接するシンボル[n-1]とシンボル[n]が同一なら1、異なるなら0とする。シンボルの値は前後のシンボルである単位トーン信号の関係だけで決定する。異なる単位トーン信号を用いた搬送波でも、単位トーン信号の周期が同じであれば同一シンボル値を得られる。

このシンボルは1ビットを1シンボルで伝送する。ボーレートは振幅信号の長さTの逆数 で求められる。単位トーン信号の長さが10msの場合、100 Baudとなる。

TBSKフレーム

デジタル値の伝送にはTBSKフレームを使用する。TBSKフレームは、TBSKmodemが信号の検出とペイロードの格納に使用するデータフォーマットである。

TBSKフレームは、信号検出と同期のためのプリアンブル部と、ビット列を格納するペイロード部で構成する。2つのフィールドは連続して存在する。

cycleの規定値4で構成したフレームレイアウトを示す。

表.TBSKフレームレイアウト

フィールド名	TBSKフレーム	
	プリアンブル	ペイロード
シンボル	14symbols	0symbols以上
シンボル値	13bits	0bits以上

TBSKフレームは、終端を明示しない。ペイロードの復調終了タイミングは、上位レイヤで定義された時間、または相関絶対値が閾値を下回る状態の継続を検出する。

隣接するフィールドにおいて、シンボル値列は独立して定義できるがシンボル列はその前段にあるフィールド終端の影響を受ける。シンボル列は差分値であり、隣接フィールドのシンボルを参照する為である。

プリアンブル部

TBSKフレームの始点を示すシンボル列である。プリアンブル部の構成長に影響を及ぼすパラメータとして、cycleを持つ。

プリアンブルは、n(bit)のシンボル値を、n+1(sym)のシンボルに展開する。差分値となるため、シンボル数が1増加する。

pythonで記述したシンボル列の生成コードは以下の定義になる。

```
b=[1]*cycle
c=[i%2 for i in range(cycle)]
d=[(1+c[-1])%2,(1+c[-1])%2,c[-1],]
preamble=[0,1]+b+[1]+c+d
```

cycleの規定値4で構成したプリアンブルのシンボル列とシンボル値列を以下に示す。

表．シンボル値(cycle=4)のプリアンブルレイアウト

No.	-1	0	1	2	3	4	5	6	7	8	9	10	11	12
サブフィールド	Prefix		Positive					Negative					Sync	
シンボル	N	P	P	P	P	P	P	N	P	N	P	N	N	P
シンボル値		0	1	1	1	1	1	0	0	0	0	0	1	0

シンボル列(cycle=4) = [N,P,P,P,P,P,P,N,P,N,P,N,N,P]
シンボル値列(cycle=4) = [0,1,1,1,1,1,1,0,0,0,0,0,1,0]

シンボル値は4つのサブフィールドがある。

Prefix　　信号出力前のトレーニング値。検出対象ではない。
Positive　　信号検出用。同位相信号の連続出力。検出、パワー測定用。
Negative　　信号検出用。反位相信号の連続出力。検出、パワー測定用。
Sync　　シンボル同期位置。シンボルの同期位置の確定用。

PositiveフィールドとNegariveフィールドはcycleパラメータにより増減する。

ペイロード部

ペイロード部は可変長のビット列である。ビット列の先頭からビット値をそのままシンボル値としてシンボルに変換する。バイト値の場合は上位ビットから順にビット値にする。

ペイロード部はプリアンブル部と連結される。ペイロードの0番目のシンボルは、前段プリアンブル部の末尾のシンボル値との差分値となる。プリアンブル部の末尾シンボルにより、ペイロード部のシンボル列は異なる。

以下は8bitsのビットストリームb00101011(0x2b)を、cycle=4のプリアンブル部に連結した様子である。cycle=4のプリアンブルは、末尾シンボルがPである点に注意する事。

表.ペイロードレイアウト

	プリアンブル	ペイロード							
bit/sym番号	-	0	1	2	3	4	5	6	7
bit値	-	0	0	1	0	1	0	1	1
sym値	0111110000010	0	0	1	0	1	0	1	1
sym	NPPPPPPNPNPNNP	N	P	P	N	N	P	P	P

TBSKmodemは1 bit/symbolであるので、ペイロード部の伝送速度は、ボーレートと同一である。

ウォームアップ/クールダウン

音響装置に出力する為に、ウォームアップとクールダウン信号を加える。TBSKフレームを変調した振幅信号の前に挿入するのがウォームアップ信号であり、後端に付加するのがクールダウン信号である。
この信号には、変調した搬送波を送信する前に、音響デバイスを定常状態にする効果がある。また、情報機器に搭載されたオーディオシステムで、音声信号の先頭と末端が正常に再生されない問題への対策として機能する。

表.信号全体のレイアウト

信号型式	非相関信号	TBSK変調信号		非相関信号
信号名	ウォームアップ	TBSKフレーム		クールダウン
		プリアンブル	ペイロード	
送信時間	0[ms]以上	T*(cycle*2+6) [ms]	N*T [ms]	0[ms]以上

ウォームアップとクールダウンに適した信号は、信号全体の範囲でシンボル長間隔で区切った波形の相関値が、連続して0に近いか、正になる信号である。
ウォームアップ信号とクールダウン信号は、プリアンブルと類似した振幅信号であってはならない。プリアンブル信号波形と類似した振幅信号とは、単位トーン信号の間隔で相関値を求めたときに、相関の絶対値が大きくなる信号である。このような信号は、プリアンブルとして誤検出される可能性が高くなる。

経験的に、ウォームアップ信号とクールダウン信号の期間は30ms程度が適当である。

完成した信号を伝送するときは、デバイスのサンプリングレートに注意し、単位トーン信号を意図した期間で出力する様に調整する。

変調復調手順

具体的なパラメータを設定して、変調復調手順について説明する。

信号の型式は、プリアンブルパラメータはcycle=4、ボーレートを80 Baud、単位トーン信号の形状はノコギリ波とする。データ部分以外に、30msのウォームアップとクールダウン信号を加える。出力データの形式は、サンプリングレート8000Hz、8bit signedのPCMデータとする。

以上の仕様で、4バイトのアスキーコード"TBSK"を伝送する振幅信号を作成する。

変調

変調処理では、ビット値列を振幅信号に変換する。以下の手順で行う。

1. トーン信号の作成
2. TBSKフレームの作成
3. シンボル列とトーン信号から変調信号を作成

オーディオデバイスなどで再生可能な音声信号とするために、次の手順を加える。

4. ウォームアップ/クールダウン信号の追加

トーン信号の生成

音源のサンプリングレートが8000Hzで80 Baudでシンボルを転送する場合、1シンボルに割り当てられるサンプル数(tick)は、8000/80=100 ticksである。

ここでは100ticksで1周期となるノコギリ波を作成する。

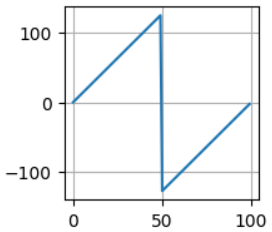


図.ノコギリ波

単位トーン信号のtick数は、経験的に50 ticksから100 ticksが適切である。短すぎると相関値の品質が悪化し、長すぎると相関値の計算コストが上昇する。サンプリング周波数毎の代表的な値を以下に示す。

表.単位トーン信号の代表値

サンプリング周波数(Hz)	単位トーン信号		ボーレート(Baud)	伝送レート(bps)
	Tick長(ticks)	信号長(ms)		
8000	100	12.50	80	80
16000	100	6.25	160	160
24000	100	1000/240 ≒ 4.17	240	240
32000	100	1000/320 ≒ 3.13	320	320
48000	100	1000/480 ≒ 2.08	480	480
8000	50	6.25	160	160
16000	50	1000/320 ≒ 3.13	320	320
24000	50	1000/480 ≒ 2.08	480	480
32000	50	1000/640 ≒ 1.56	640	640
48000	50	1000/960 ≒ 1.04	960	960

TBSKフレームの作成

この仕様のTBSKフレームは、プリアンブル部=14 symbols、ペイロード部=32 symbolsの合計46 symbolsである。

先にプリアンブル部のシンボル列を作成する。仕様により、cycle=4のプリアンブル部のシンボル列は[NPPPPPPNPNPNNP] になる。

次にペイロードのシンボル列を作成する。文字列TBSKを一文字づつASCIIコードのビット値にし、プリアンブル部の末尾がPであることを考慮し、ビット列を差動値のシンボル列に変換する。

表.TBSKを格納するペイロードのレイアウト

文字	T	B	S	K
ASCIIコード	86	66	8.	75
ビット表現	01010100	01000010	01010011	01001011
シンボル列	NNPPNNPN	PPNPNPPN	PPNNPNNN	PPNPPNNN

シンボル列全体	NNPPNNPNPPNPNPPNPPNNPNNNPPNPPNNN

作成した2つのシンボル列を結合して、TBSKフレームのシンボル列が完成する。

表.TBSKを格納するTBSKフレームのレイアウト

フィールド名	プリアンブル	ペイロード
シンボル列	NPPPPPPNPNPNNP	NNPPNNPNPPNPNPPNPPNNPNNNPPNPPNNN
シンボル列全体	NPPPPPPNPNPNNPNNPPNNPNPPNPNPPNPPNNPNNNPPNPPNNN	

シンボル列とトーン信号から変調信号を作成

TBSKの変調信号は、シンボル列の値を元に、単位トーン信号を正転又は反転して連結したものである。TBSKフレームのシンボル列の先頭から、シンボルがPならそのまま、Nなら反転した単位トーン信号を順番に連結していく。

この結果、46 symbolsのシンボル列は、4600 ticksのPCMデータに展開される。

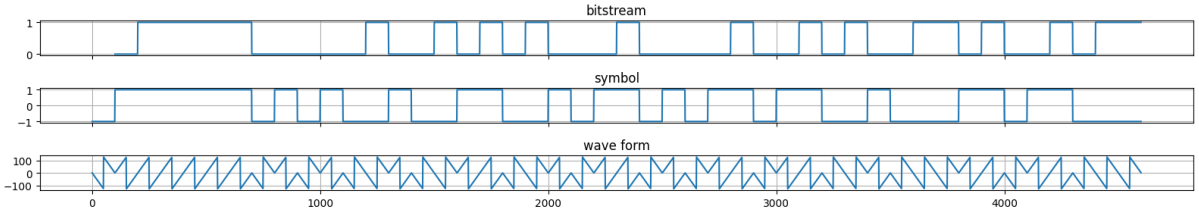


図.展開前の値と展開後のPCM

ウォームアップ、クールダウン信号の作成

ウォームアップ、クールダウン信号は、相関性の無い信号が望ましい。簡単な生成方法として、乱数値が利用できる。ここでは、XorShiftで生成する。240個のUINT8値を生成し、signdに変換し、0点を調整する。

信号は以下のPythonコードで生成できる。

```
""" https://ja.wikipedia.org/wiki/Xorshift
"""
def xorshift31(size=10,seed=299,skip=0):
    ret=[]
    y=seed
    for i in range(size):
      y = y ^ (y << 13)
      y = y ^ (y >> 17)
      y = y ^ (y << 5)
      y=y & 0x7fffffff
      ret.append(y)
    return ret
v=[(i%256)-128 for i in xorshift31(size=240)]
```

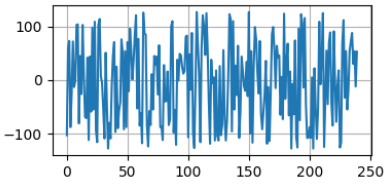


図.無相関信号

完成したTBSKmodem信号

変調信号の前後にウォームアップ信号とクールダウン信号を付け加えて、TBSKmodemの信号が完成する。以下は完成した信号全体の振幅波形である。

合計で5080 ticks 、8000 Hzサンプリングで635 msの信号となる。

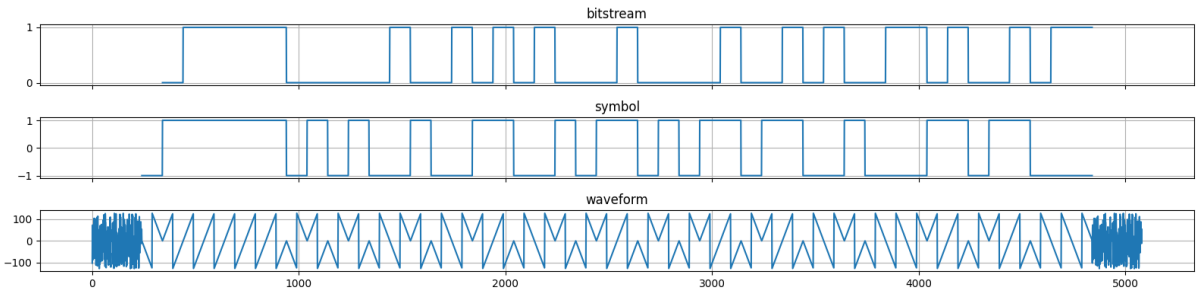


図.シンボル、変調信号

復調

復調処理では、振幅信号をビット値列へ変換する。以下の手順で行う。

1. 相関値分析
2. Positiveサブフィールドの検出
3. Negativeサブフィールドの検出
4. 同期位置の決定
5. プリアンブルの評価
6. ペイロードの取得
7. 終端の推定

これらの処理は逐次実行することが望ましいが、簡単のため、記録した振幅信号を処理する方式を説明する。復調元の振幅信号は、変調で作成した信号を使う。

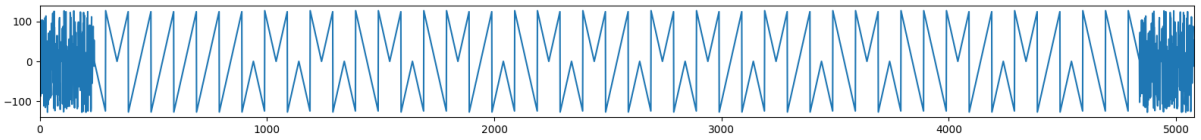


図.復調元の信号

相関値分析

振幅信号からシンボルを求めるには、振幅信号(A)から振幅信号を、シンボル長(100 ticks)だけ遅延させた信号Bを作成し、振幅信号全体について、シンボル長の期間で相関値を連続して計算する。

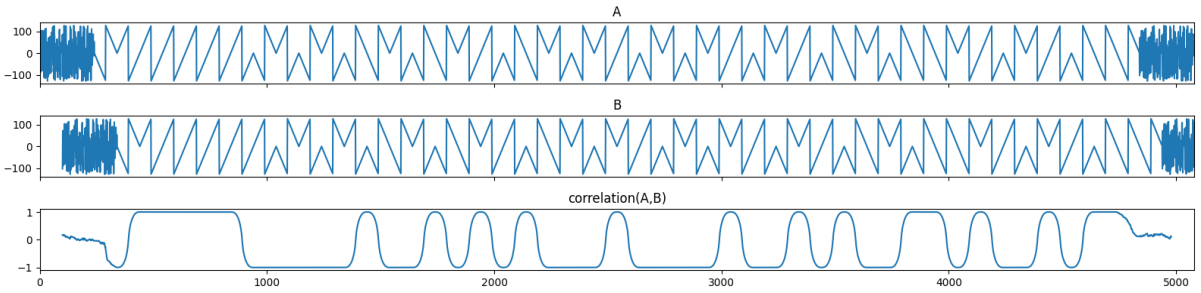


図.振幅信号の相関値

計算した相関値は、[-1,1]の範囲を持つ時系列値である。

Positiveサブフィールドの検出

振幅信号の先頭から、相関値が連続して高い箇所を探索する。これはPositiveサブフィールドの部分に相当する。

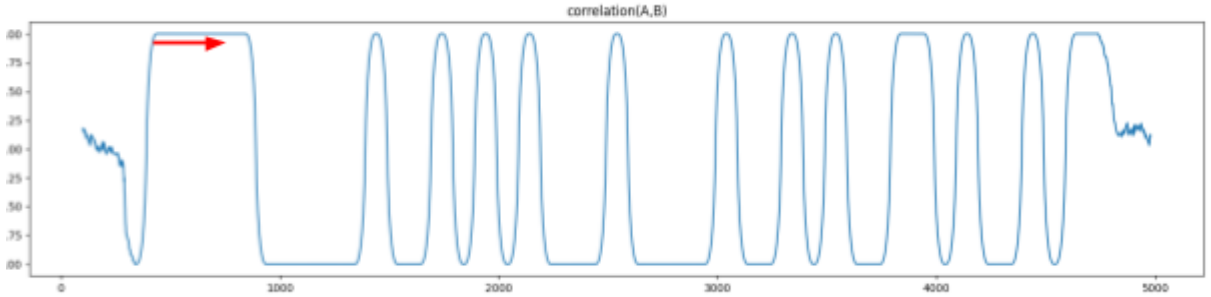


図.Positiveサブフィールドの検出箇所

Negativeサブフィールドの検出

次に、Negativeサブフィールドの開始点を探索する。
Positiveサブフィールドに続いてNegativeサブフィールドが存在する場合、相関値は大きなギャップを伴って低下する。そのギャップが閾値を超えた時点で、Negativeサブフィールドが存在すると判定する。

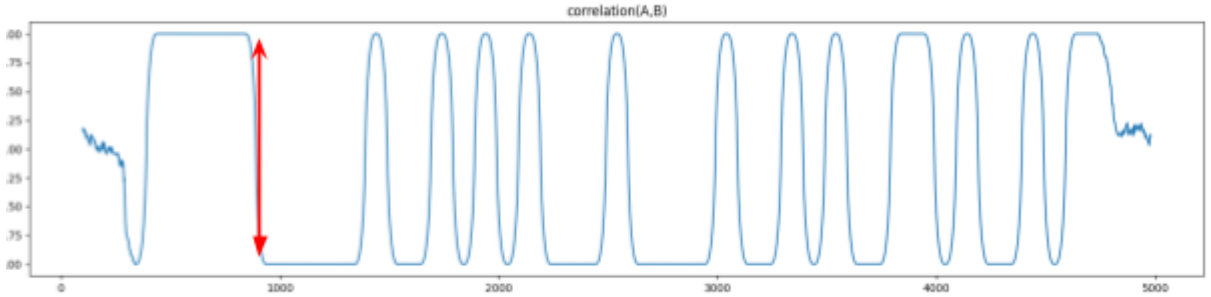


図.Negativeサブフィールドの検出箇所

同期位置の確定

Negativeサブフィールドに続くSyncサブフィールドに到達すると、相関値が正の値を取り、ピークを形成して再び負になる。このピークがシンボル境界のエッジであり、信号の同期位置となる。

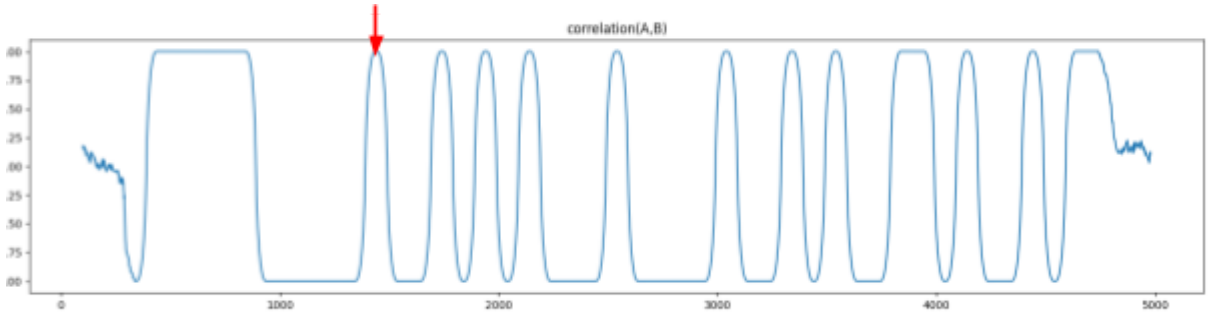


図.同期位置の検出箇所

プリアンブルの評価

同期位置を基準に、Positiveサブフィールドと Negativeサブフィールドの品質を評価する。各サブフィールドの区間において、規定の期間と相関値が閾値範囲内にあることを確認する。品質が良好であれば、信号が存在すると見なしてペイロードの取得に進む。そうでない場合は、Positiveフィールドの検出からやりなおす。

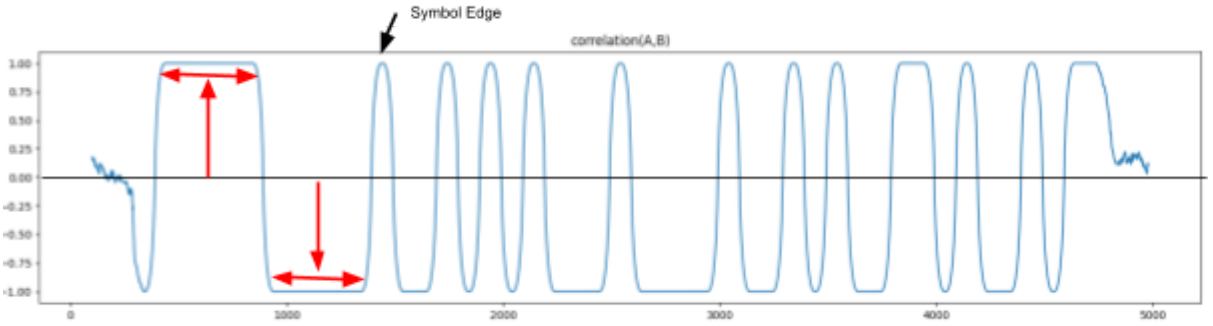


図.プリアンブルの評価箇所

ペイロードの取得

同期位置を基準にしてプリアンブルに続く相関値をシンボル長間隔で取得すると、相関値からシンボル値を得ることができる。シンボル値を8ビット単位で区切ってASCIIコードを復元する。

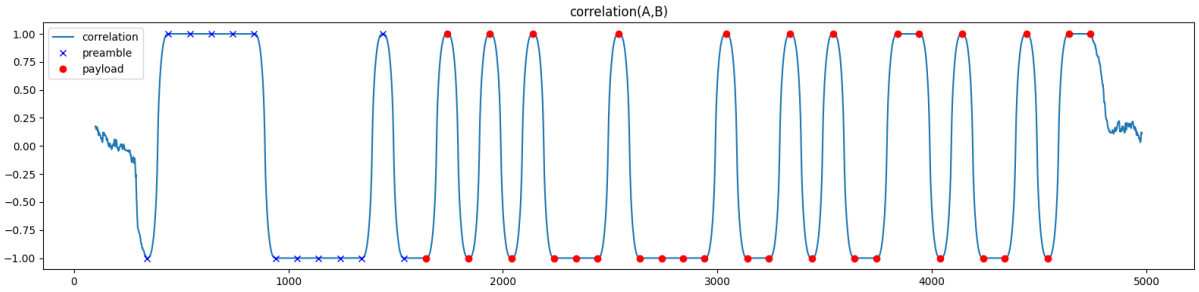


図.シンボル値の読取位置と相関値

精密に調整されていないオーディオ装置では、AD/DA変換に僅かな周波数ずれや変動があり、伝送が長時間(概ね10k ticks以上)になるとシンボル境界が一致しなくなる。これに対応するため、ペイロードの受信中に同期位置を修正する必要がある。

ペイロードのシンボル値が反転するとき、相関値には勾配が発生する。この時、より相関値の高いほうに同期点をシフトすることで、同期点を追跡することができる。

正し、この方式はシンボルの反転しない信号では意味をなさない。より強力に作用させるならば、上位レイヤでエンコーディングが必要である。

終端の推定

振幅信号が有限である場合、終端までシンボルを読み取ることで終端を確定できる。無限である場合、相関値の絶対値が閾値を下回った時点で終端と仮定する。
クールダウン信号に無相関信号を使用することで、終端の確定を促進することができる。ただし雑音に対して脆弱である。終端は上位プロトコルで規定するべきである。

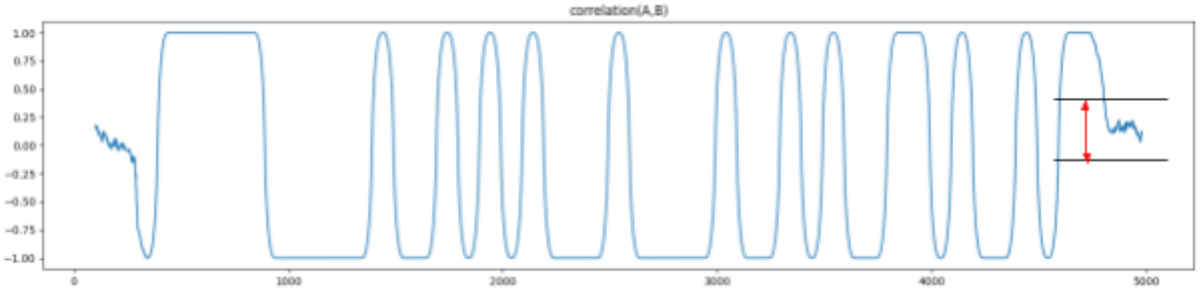


図.終端位置の推定

性能評価

主観による評価である。

通信速度

通信速度は、単位トーン信号を100 ticks以上で構成すると良好な結果になる場合が多い。安定した通信環境であれば、50 ticksまでは正常な通信が確認できた。

100ticksで構成した単位トーン信号の場合、サンプリングレートが8kHzの場合80 bps、16kHzであれば160 bpsの通信速度が期待できる。外付けマイクとスピーカーの通信では、48kHz/50ticksの設定で、60cmの距離で960bpsの通信に成功した。

通信距離

160 bpsの場合、通信距離は数m程度である。基本的には音響デバイスの再現性と、通信環境による。反響の多い室内では、1m程度の通信距離である。

環境特性

PC、スマートフォンに搭載されているマイク、スピーカー、外付けスピーカー、何れのデバイスを介しても通信できた。また、インターネットを使用する音声通信アプリを中継しても、通信が成立した。
組込機器の場合、圧電ブザー単体による送受信も不可能ではないが、受信については適切なアナログ回路を用意する必要がある。

ノイズ特性

通信路の空間ノイズ耐性は一般的なPSK通信と同等です。周波数帯域の一部に多少のノイズが混入しても問題はない。スペクトラム拡散したトーン信号の場合は、伝送路で一部の周波数帯域が消失しても問題はない。

エラー特性

音楽の録音再生性能がある音響デバイスならば、生活音程度のノイズではエラー発生しません。ランダムノイズはビット反転エラーとなり修復される場合もあるが、バーストエラーは同期位置を見失い、後続のデータが全て破壊され続ける。

実装

TBSKmodemの仕様を実装した各言語向けのライブラリとその特徴を示す。全てのライブラリは通信仕様に互換性があり、相互に信号を変調し、復調することができる。

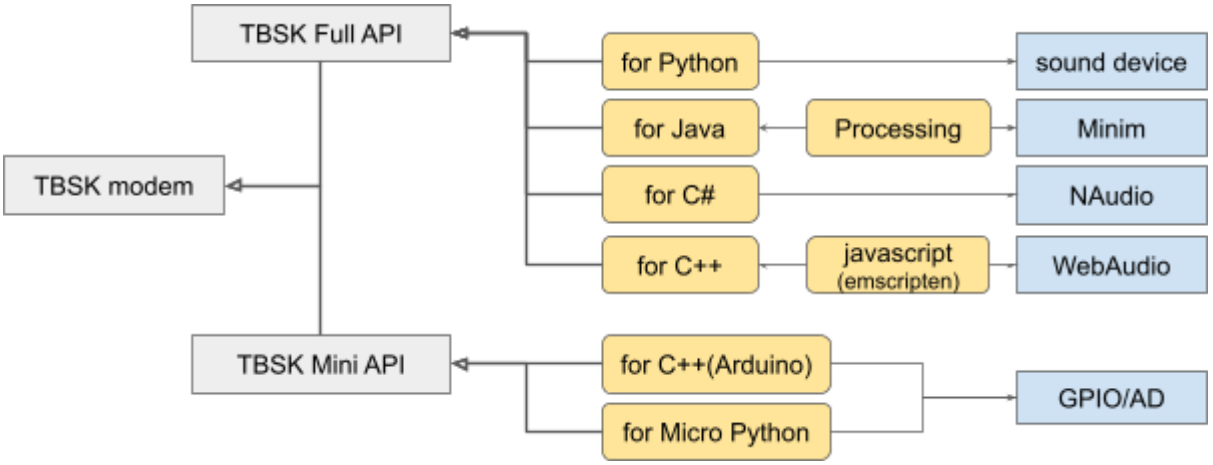


図.ソフトウェア群の構成

コア実装

コア実装は、複変調演算処理の実装を持つコードセットである。設計思想の異なるFullとMiniの系統があり、それぞれ多言語での実装がある。Fullはメモリや演算力が豊富なシステム向けの実装である。Miniは、メモリや演算力が制限されたシステム向けの実装で、整数演算のみを使用する。どちらの実装も、標準ライブラリ以外の外部ライブラリには依存しない。

Python

Full系統。Full系統のリファレンス実装である。
開発環境はAnaconda環境を推奨する。
オーディオライブラリはsounddeviceを利用できる。

https://github.com/nyatla/TBSKmodem

Java

Full系統。マネージド言語のリファレンス実装である。APIはPython版を踏襲する。
演算実装のみ。オーディオライブラリには対応しない。

https://github.com/nyatla/TBSKmodemJava

C#

Full系統。純粋なC#の実装である。APIはJava版を踏襲する。
オーディオライブラリはNAudioを利用できる。

https://github.com/nyatla/TBSKmodemCS

C++

Full系統。C++の実装である。APIはJava版を踏襲する。
演算実装のみ。オーディオライブラリには対応しない。
VisualStadio、Linux、Emscriptenのビルド環境がある。
https://github.com/nyatla/TBSKmodemCpp

Micro

Mini系統。C++とMicroPythonがある。搬送波には矩形波を使用する。APIは同期式で他の実装と異なる。

MicroPythonの実装は、変調のみの実装である。
C++の実装はArudino環境で使用できる。メモリ管理機構は使用しない。
変調・復調が可能。出力にGPIO、入力にADピンを使う。
RaspberryPi picoで動作実績がある。

https://github.com/nyatla/TBSKmodemMicro

ラップライブラリ

コア実装をラップして、環境に適したAPIを整備した実装である。

Processing

Java版をラップした実装である。Processing標準のSerialライクなAPIと、専用のModemAPIを提供する。
オーディオライブラリはminimを使用する。

https://github.com/nyatla/TBSKmodem-for-Processing

Javascript

C++版をEmscriptenでコンパイルしてJavascriptAPIを整備した実装である。
WebsocketライクなAPIを提供する。スタンドアロン、npmに対応する。
出力にはWebRTC/wasm/WebAudioを使用する。

https://github.com/nyatla/TBSKmodemJS

ライブデモ

Javascript版を用いたライブデモである。
Chrome/Edge/Safariなどのモダンブラウザで動作する。

https://nyatla.jp/tbskmodem/