通信システムを実装するには？

佐波　孝彦

コミュニケーション

- **人間が互いに意思・感情・思考を伝達し合うこと。言語・文字その他視覚・聴覚に訴える身振り・表情・声などの手段によって行う。**
- **通信（Communications）とは？**

① 意思や様子などを他人に伝えること。音信を通じること。信書をやりとりすること。たより。

② 郵便・電信・電話・信号・パソコンなどを使って意思や情報を伝達すること。

（出典：スーパー大辞林）

コミュニケーションに必要なもの

- **「夥多」を読めますか？**
  - 「かた」＝多すぎること，夥しい（おびただしい）
- **「circumlocution」の意味がわかりますか？**
  - 遠回しな表現、婉曲表現

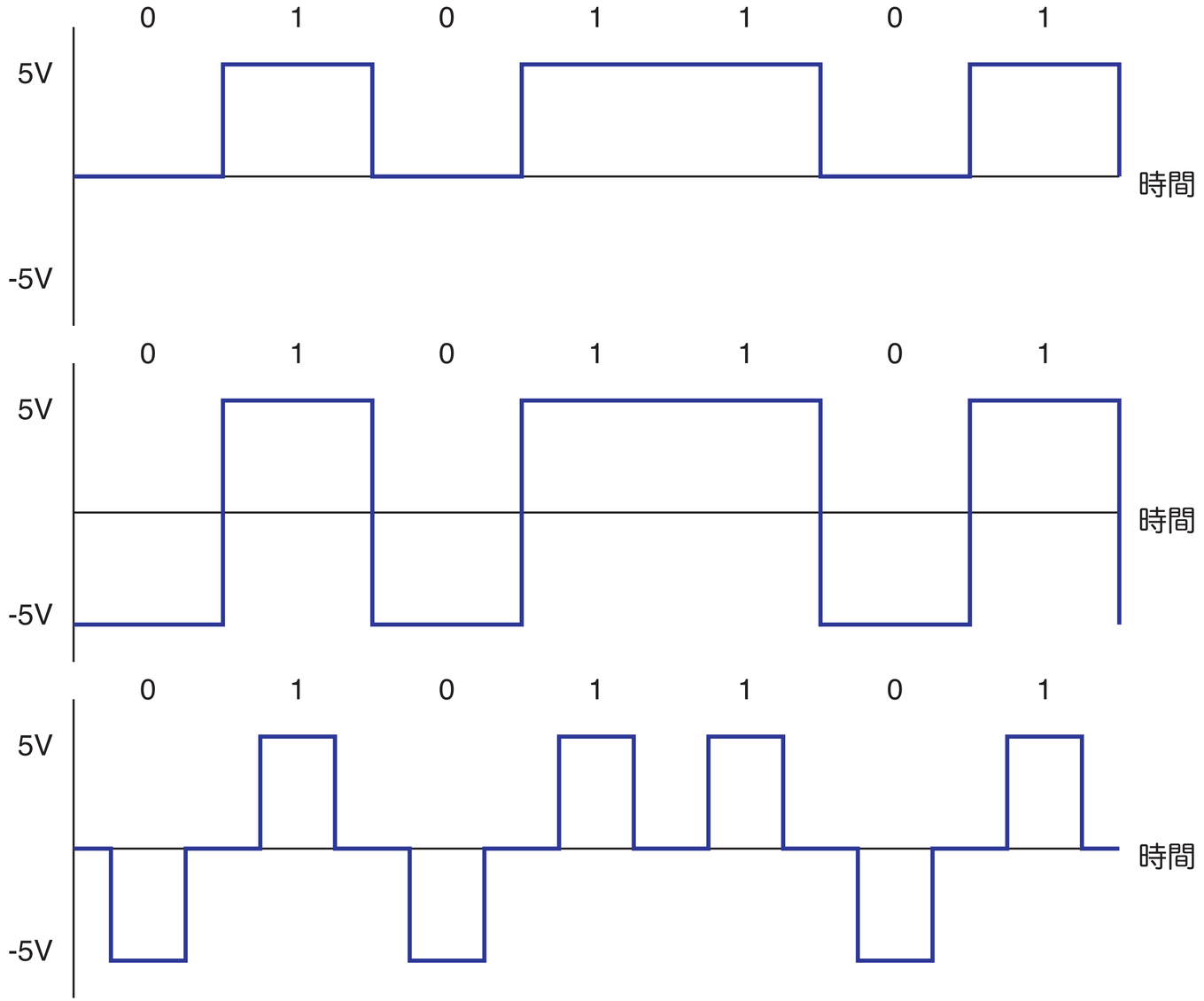
お互いが言葉を知らなければ
コミュニケーション（通信）は成立しない

どのような言葉が必要？

- **ディジタルデータを信号で表すには？**
  - バイナリ信号

    → 0, 1の2つの状態を識別できる信号
- **プロトコル**
  - 信号送信の手順，データの表現法など，データ通信を行うために予め決めておく規約

バイナリ（2値）信号の例

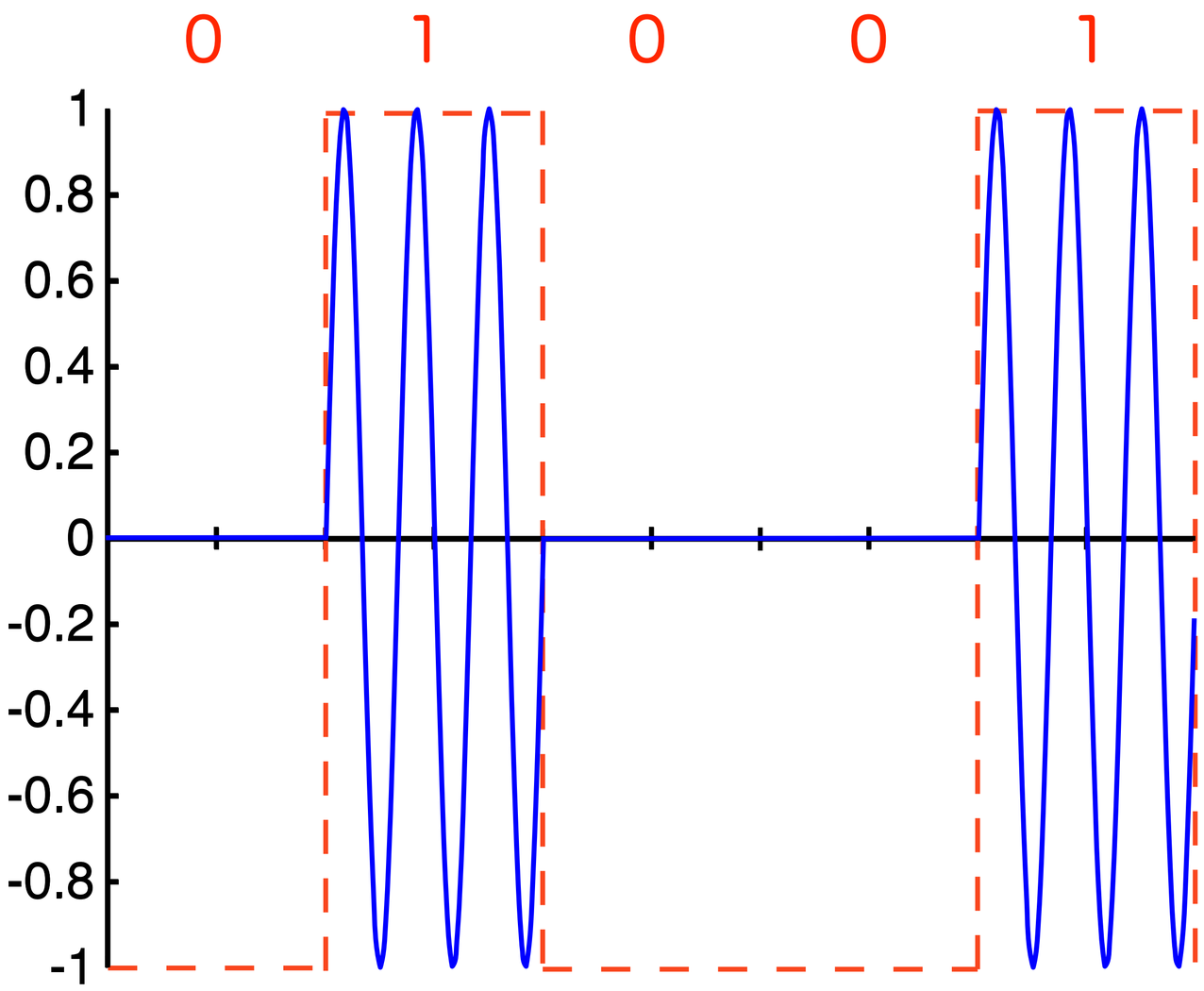


信号のオン・オフを識別（NRZ: Non-Return-to-Zero）

正負判定で識別（NRZ: Non-Return-to-Zero）

0や1が連続してもビットを識別可能（RZ: Return-to-Zero）

バイナリ（2値）信号の例



正弦波のオンとオフで1と0の状態を表している

リモコン（光通信）などはこのタイプ

モールス信号も同様の波形

基本となる正弦波の波形の形状を変化させることを変調という

波形からビット列を復元することは復調という

マルチレベル信号

- **識別が可能なら状態数は2値（2状態）でなくとも良い**
  - ▶4値（2ビット）
  - ▶8値（3ビット）
  - ▶16値（4ビット）
  - ▶32値（5ビット）
- **1つの波形で送れるビット数が増えるので速度を稼げる**
  - ▶ただし，多値数が増えると，判定誤りが起きやすくなる

パルス変調

- **識別対象は電圧（振幅）でなくとも良い**
  - ▶パルス振幅変調（PAM: pulse amplitude modulation）
  - ▶パルス幅変調（PWM: pulse width modulation）
  - ▶パルス密度変調（PDM: pulse density modulation）
  - ▶パルス位置変調（PPM: pulse position modulation）

多値化する場合，複数のパラメータで状態数を

増やしても良い（振幅と位置の組み合わせで4状態等）

無線伝送メディア

- **電磁波（電波），** ただし電波を出すには免許（認可）が必要
  - ▶技適マークのある認証済み装置（Wi-FiやBluetooth等）は免許なしでも使えるが，今回のテーマでは使用不可
- **光（可視光，赤外線などの不可視光線）**
- **音（可聴音，超音波）**

チーム独自のプロトコルを実装する

通信システム構築で考えること

- **目的と要求事項の明確化**
  - どのようなデータを，どのくらいの速度で，どの程度の遅延で，誰とやり取りしたいのかを明確にする.
- **物理層の決定**
  - 通信に用いる媒体（無線，有線，光ファイバなど）を決める.
  - 伝送速度や帯域幅を決定する.

通信システム構築で考えること

- **目的と要求事項の明確化**
  - どのようなデータを，どのくらいの速度で，どの程度の遅延で，誰とやり取りしたいのかを明確にする.
- **物理層の決定**
  - 通信に用いる媒体（無線，有線，光ファイバなど）を決める.
  - 伝送速度や帯域幅を決定する.

計算機でのテキストの表現

- **文字をビット列で表す（符号化する）必要がある**
  - アルファベットなら26文字
  - 平仮名なら46文字
  - 文章を書くなら，句点（ピリオド），読点（カンマ），空白文字，改行などの文字も必要
  - カタカナ・漢字まで入れると膨大な種類

文字符号化の例

- ASCII - 7 bit（英文）
- ISO - 8 bit（欧文）
- JIS (ISO-2022-JP) - 16 bit（日本語）
- シフトJIS (Shift_JIS) - 16 bit（日本語）
- EUC (EUC-JP) - 16 bit（日本語）
- UTF-8, UTF-16, UTF-32 - 16, 32 bit（各国語）

ASCIIコード

- 2進数で“1000111”なら“G”, “1101101”なら“m”を表す（16進数なら47と6D）
- 改行コード（制御文字）
  - LF = Line Feed（UNIX系OS, macOS）
  - CR = Carriage Return（昔のMac OS）
  - CR+LF (Windows系OS)
- SOH = start of heading, STX = start of text 等，略語は各自で調べてください

ASCIIコード

			上位ビット							
	Binary		000	001	010	011	100	101	110	111
		Hex	0	1	2	3	4	5	6	7
下位ビット	0000	0	NUL	DLE	SP	0	@	P	`	p
	0001	1	SOH	DC1	!	1	A	Q	a	q
	0010	2	STX	DC2	"	2	B	R	b	r
	0011	3	ETX	DC3	#	3	C	S	c	s
	0100	4	EOT	DC4	$	4	D	T	d	t
	0101	5	ENQ	NAK	%	5	E	U	e	u
	0110	6	ACK	SYN	&	6	F	V	f	v
	0111	7	BEL	ETB	’	7	G	W	g	w
	1000	8	BS	CAN	(	8	H	X	h	x
	1001	9	HT	EM	)	9	I	Y	i	y
	1010	A	LF	SUB	*	:	J	Z	j	z
	1011	B	VT	ESC	+	;	K	[	k	{
	1100	C	FF	FS	,	<	L	\	l	|
	1101	D	CR	GS	-	=	M	]	m	}
	1110	E	SO	RS	.	>	N	^	n	~
	1111	F	SI	US	/	?	O	_	o	DEL

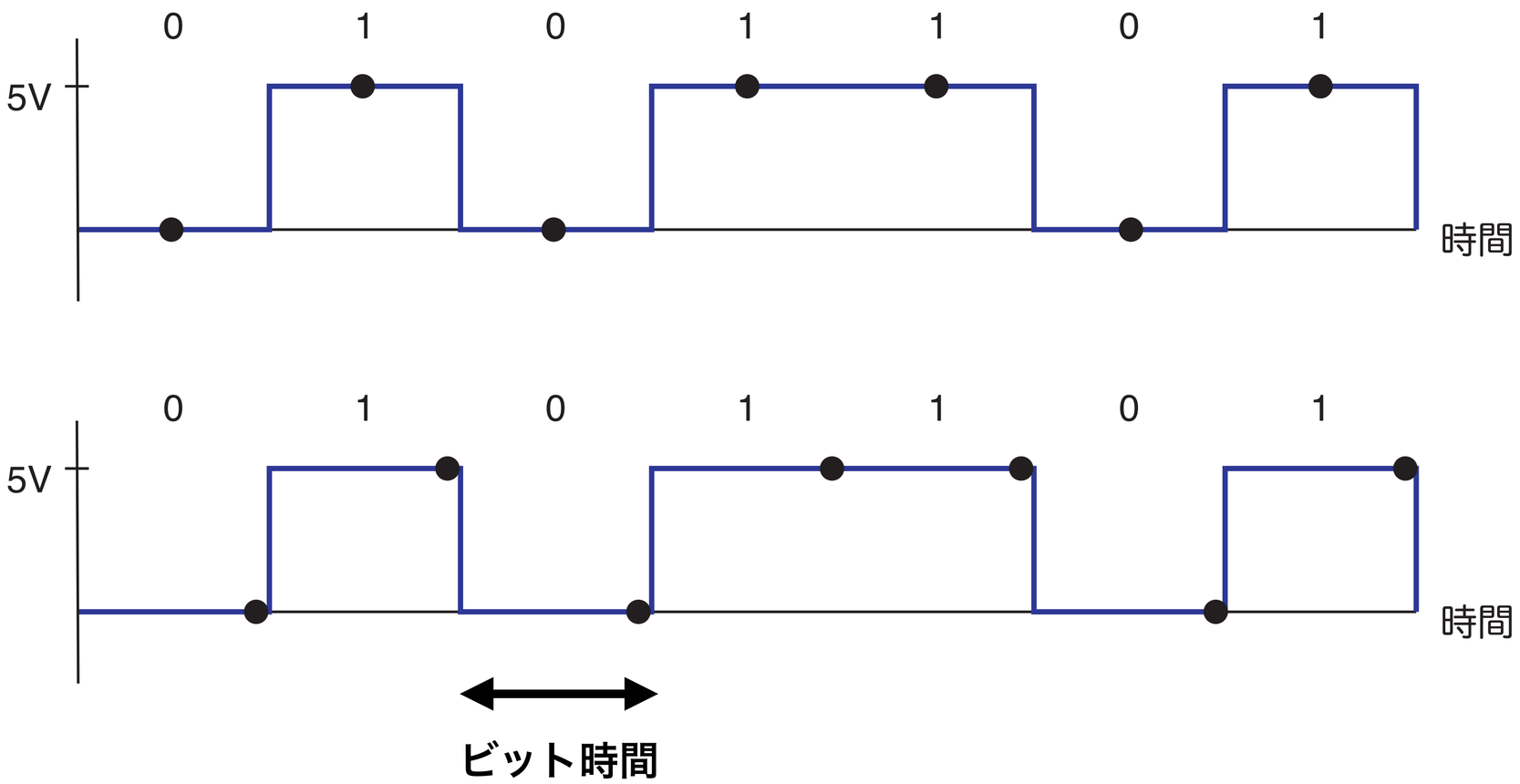
通信システム構築で考えること

- **目的と要求事項の明確化**
  - どのようなデータを，どのくらいの速度で，どの程度の遅延で，誰とやり取りしたいのかを明確にする.
- **物理層の決定**
  - 通信に用いる媒体（無線，有線，光ファイバなど）を決める.
  - 伝送速度や帯域幅を決定する.

Arduinoでの信号受信

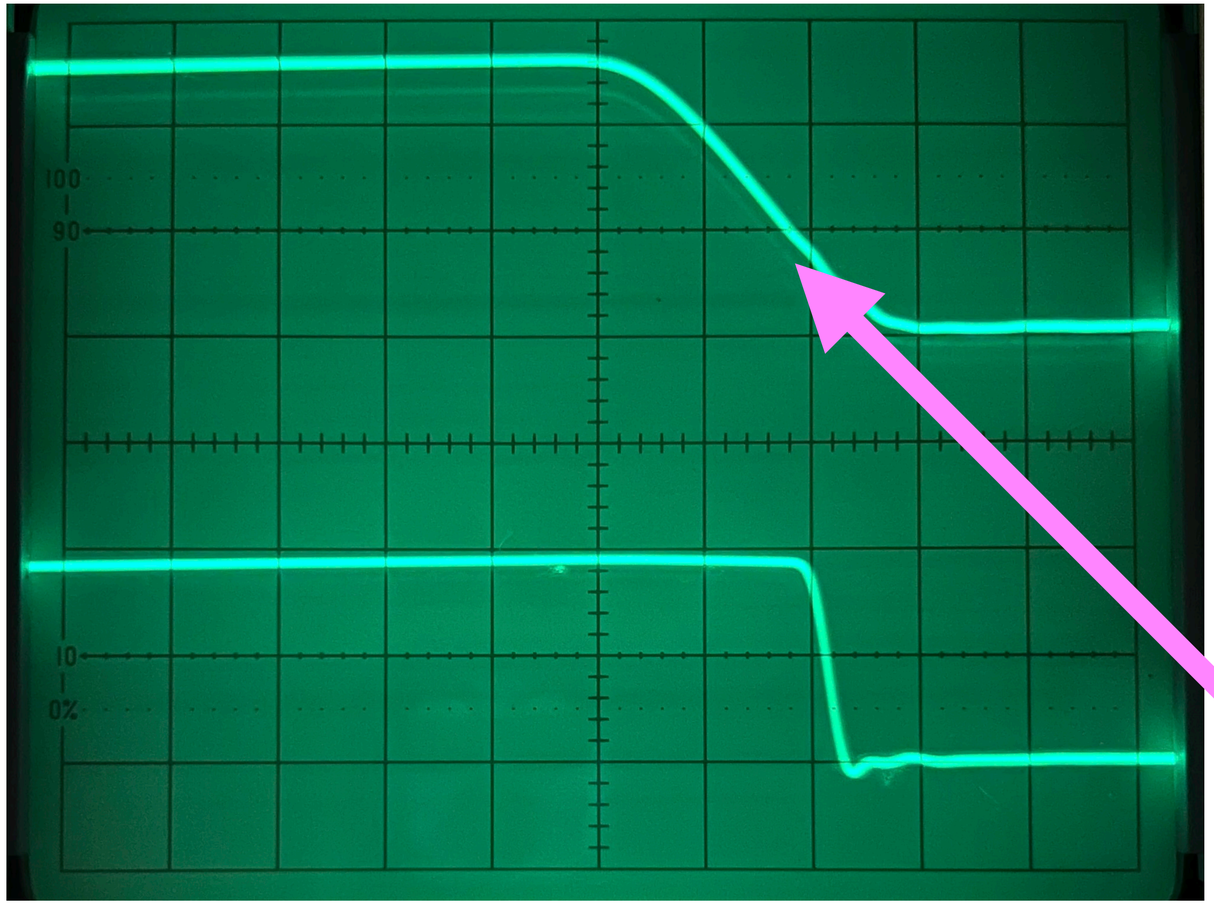
- **アナログリード**
  - analogRead()関数で，0V-5V（or 3V or 参照電圧）を0-1023（10 bit）の値で返す．サンプリング周期は？
- **デジタルリード**
  - digitalRead()関数が読み込まれたとき，High（5V）かLow（0V）を読み取る．

読み取りのタイミング



- ビット時間の周期で正しく読み取れるなら判別可能
- しかし，タイミングの制御は難しいし，揺らぎもある

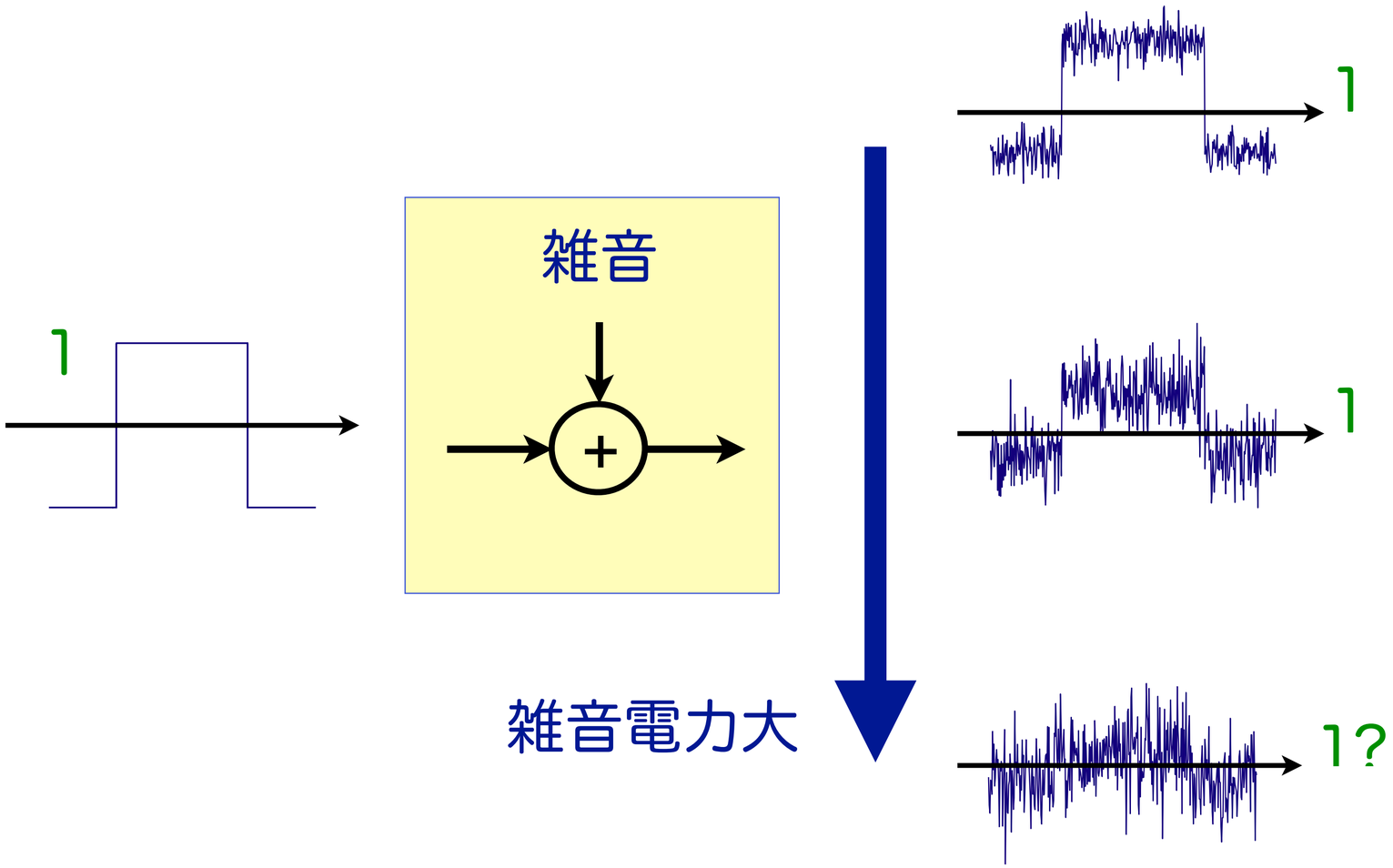
実際のパルス波形（拡大）



実際のパルス波形は拡大すると，なだらかな変化

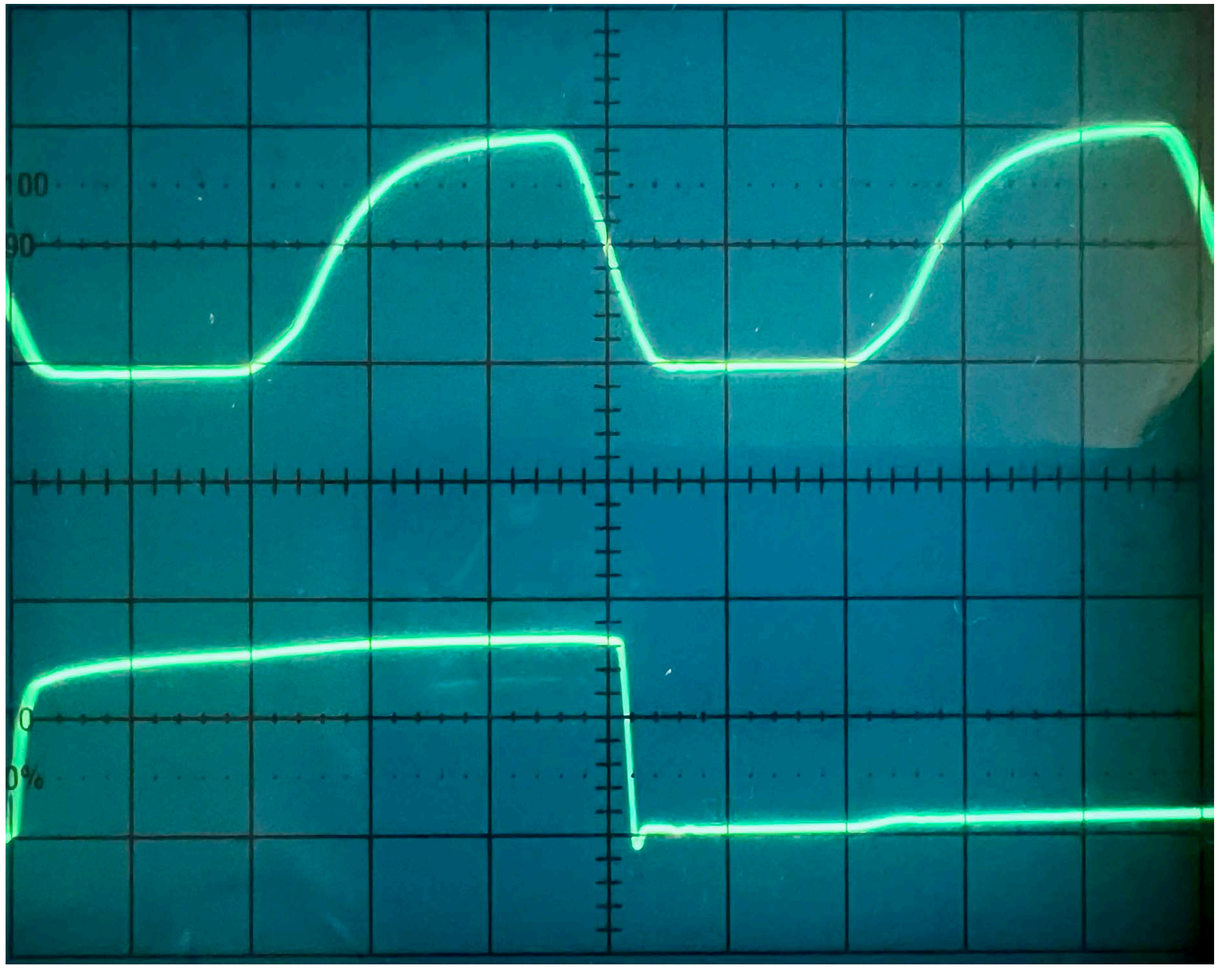
読み取りタイミングがここだったら誤判定？

そもそも受信波形が歪んでいる場合も



- **雑音の要因**
  - 電波: 電磁ノイズ
  - 光: 背景光，強度雑音
  - 音: 背景音

ビット時間を短くすると



判定は1パルスあたり1点で良いか？（雑音で，たまたま振幅が小さかったら？）

→平均や積分の利用？

色々考慮してビット時間を決める必要がある

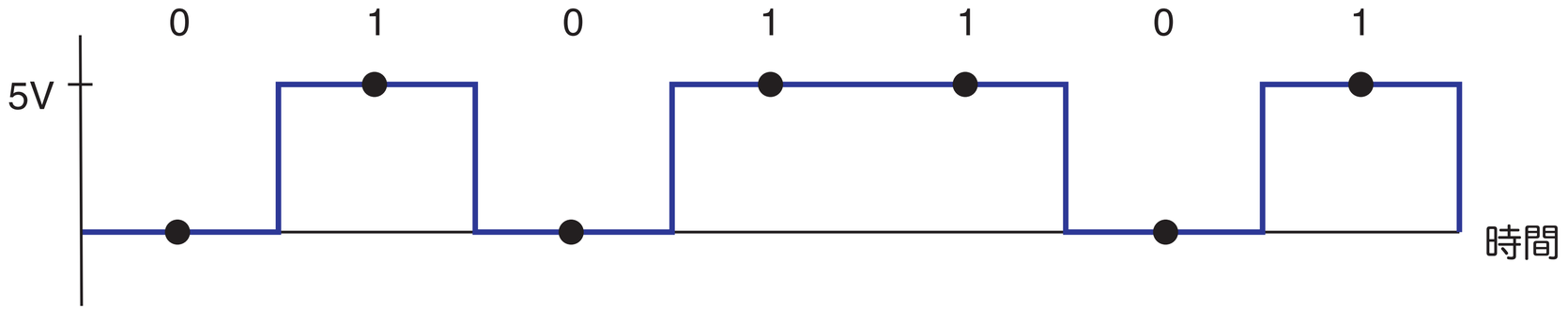
通信システム構築で考えること

- **データリンク層の決定:**
  - フレームの形式（ヘッダ，データ，フッタなど）を決める.
  - アドレス指定方法（誰宛？）を決める.
  - エラー検出・訂正方法を決める.

} 今回は考えなくとも良いかも（考えても良い）

どこからデコード？

仮に正しく復調（ビット判定）ができたとしてもデコード開始位置を間違えると．．



“0101101”は，ASCIIコードの場合，記号の「ー（マイナス）」

どう対処する？

最初の”0”を見逃して，“1011010”と判定したら，アルファベットの「Z」

モールス符号の場合

- **モールス符号とは**
  - 0と1を短点（トン）と長点（ツー）で表す文字符号
  - 音（正弦波）を使った パルス幅変調とも言える
  - 低品質な回線でも聴き取ることが可能なため，遠洋航海での通信では通信衛星が普及し出す1999年頃まで使われていた

モールス符号の例

欧文		和文		数字	
.-	A	-.-	ア	.----	1
-...	B	.-	イ	..---	2
-.-.	C	..-	ウ	...--	3
-..	D	-.---	エ	....-	4
.	E	.-...	オ	.....	5
..-.	F	.-..	カ	-....	6
--.	G	-.-..	キ	--...	7
....	H	...-	ク	---..	8
..	I	-.--	ケ	----.	9
.---	J	----	コ	-----	0

文字や単語の区別は？

I AM

•• •– ––

符号間 単語間 文字間

間隔を変える

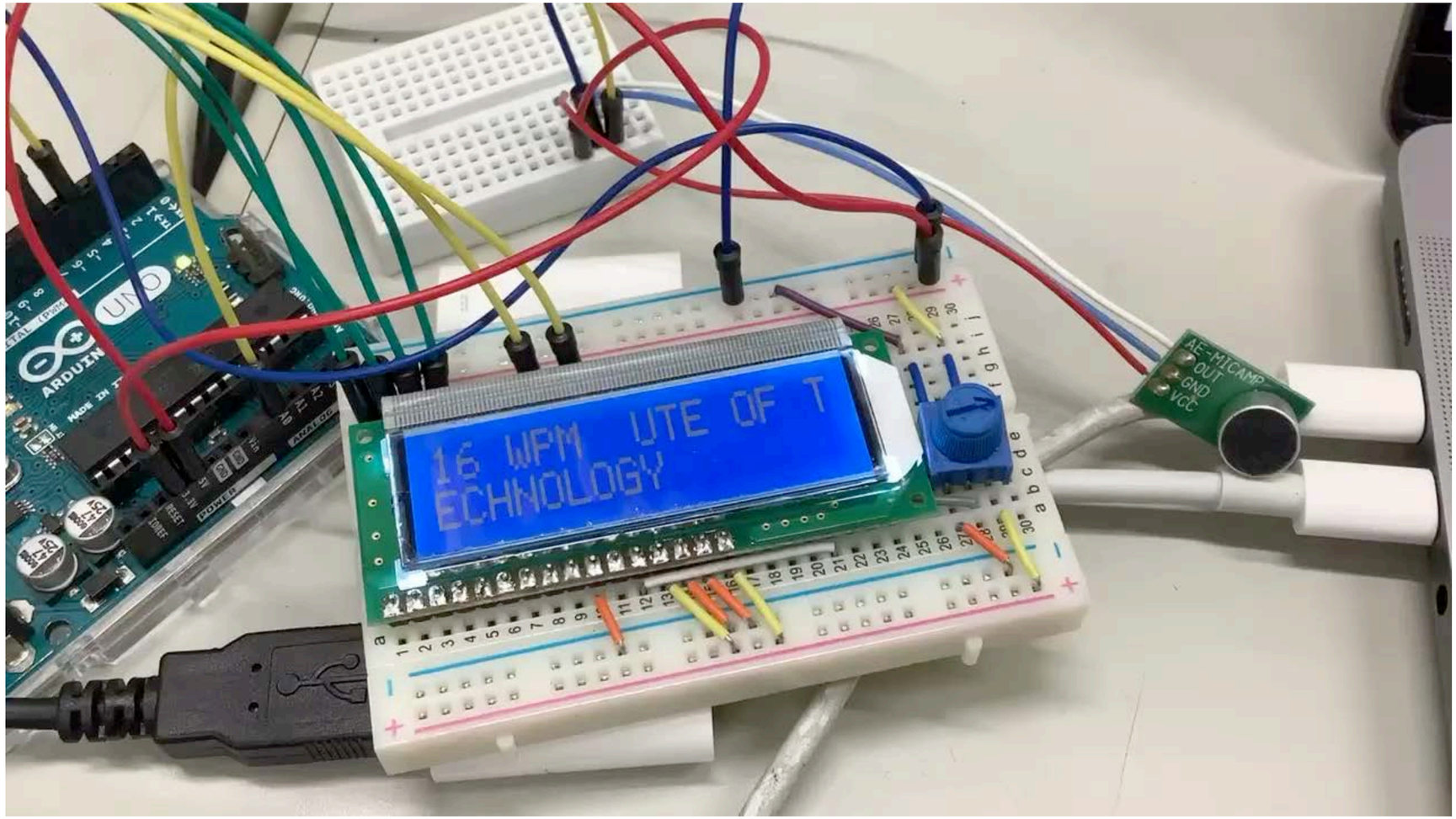
	長さの比率
短点	1
長点	3
符号間	1
文字間	3
単語間	7

人によって速度はまちまち

「ホレ」と「ラタ」

- **「ホレ」（ー・・ーーー）で、和文（送信）を始めることを表す。**
  - ▶元々、電報の宛先と本文を分ける時に使っていた符号。
- **「ラタ」（・・・ー・）で和文（送信）が終了することを表す。**
  - ▶和文の中で欧文を使う時は（　）でくくる。

モールス受信機実装例



PCのキーボードの場合

- **キーボードのキーで直接文字コードのビット列を送っているわけではない**
  - キーボードのキー数は標準で109なので，7 bit コードでも1対1の対応は無理
  - キーの組み合わせ（大文字はシフトと組み合わせるなど）を使っても 8 bit コードだと足りない
  - OSやIME（Input Method Editor）がソフトウェア上で対象文字コードに変換

キーボードからの信号

- **キースイッチが押されるとMakeコード（パルス列）を送信**
- **キースイッチのデータを送信**
- **離された時にBreakコード（パルス列）を送信**
  - キーを押した時と離したときで異なる情報を送ることで，押されたキーを知らせる
  - 1つのデータは11ビットで構成

    データの開始位置（あるいは終了位置）を知らせるための符号列を使う（ヘッダやフッタと呼ばれる）

物理層との関係

- **7 bit のASCIIコードを4値信号で送る場合**
  - “0101101”は，”01”，”01”，”10”，”1＋(0 or 1)”に分けて1波形にする必要がある．
  - 文字の符号を，連続で送ればいいわけでは無い（文字の判別のため）
  - パケット化するには，物理層の信号にもヘッダ，フッタを加えるのが普通

処理の流れ（1対1の場合）

- Macでテキストを入力（送信データはテキスト）
- 文字をビット列に変換し，ヘッダやフッタを付加
- パケットを識別するためのヘッダやフッタを付加
- データを使って信号波形を生成（変調）し伝送
- 受信信号を復調しビット列を再生
- ヘッダやフッタを元に文字のデコード
- Macの画面あるいはLCDに表示

まずは物理層の媒体を決めてください

- 音，超音波
- 糸電話や振動センサ（接触が必要ですがこういうのは許可）
- 光（LED＋照度センサ）
- 色情報（カラーセンサ）
- ホールセンサ（磁気センサ）
- 扇風機（モーターの回転数計測）
- 他にも色々な可能性があります

ハッカソンとしての評価

- **評価項目：**
  - 正確な伝送および速さ
  - 技術的な工夫及び独創的な点
  - エンタメ要素も歓迎（加点判定）
- **優勝チームは表彰も？**

本日の演習

初めに，Arduino　IDE　のシリアルモニターから文字列を入力し，改行が入力されるまで受け付け，それぞれの文字をASCIIコード（8 bitの2進数）に変換し，結果を

文字 -> ASCIIコード

の形式で出力するプログラムを考えます.

セットアップ部分

```
void setup() {
  Serial.begin(9600); // シリアル通信を開始
  Serial.println("文字を入力してください:");
}
```

ループ部分

```
void loop() {
  if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n'); // 改行まで読み込む
    Serial.println("ASCIIコード（2進数）:");

    for (int i = 0; i < input.length(); i++) {
      char c = input.charAt(i);
      Serial.print(c);
      Serial.print(" -> ");
      printBinary((byte)c);  // 2進数表示
      Serial.println();
    }
   Serial.println("-----");
  }
}
```

独自関数の定義

```
// バイトを2進数形式で表示する関数
void printBinary(byte b) {
  for (int i = 7; i >= 0; i--) {
    Serial.print(bitRead(b, i));
  }
}
```

課題：4進数で表示するには？

- 4値伝送を考える場合には，4進数（0~3）で表示できるようにすることも必要
- 桁数は揃えた方が良いはず（4桁→0000~3333）

ループ部分

```
void loop() {
  if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n'); // 改行まで読み込む
    Serial.println("ASCIIコード（4桁の4進数）:");

    for (int i = 0; i < input.length(); i++) {
      char c = input.charAt(i);
      byte ascii = (byte)c;
      Serial.print(c);
      Serial.print(" -> ");
      printBase4(ascii);  // 4桁で表示→ここを考えるのが課題
      Serial.println();
    }
    Serial.println("-----");
  }
}
```