訂正_(第1回1. Arduino)



第3回 Arduino 入門 シリアル通信・パソコン通信編

プレゼン by いっちー

目次

- 1. シリアル通信とは
- 2. マイコンでのシリアル通信例
- 3. 通信速度の設定がずれると・・・
- 4. どうやって PC とつなぐの?
- 5. シリアル- USB 変換 IC
- 6. PC にデータを送信してみよう
- 7. ASCIIコード
- 8. シリアルモニタ
- 9. スイッチ入力プログラム
- 10. シリアルモニタの表示
- 11. PC から受信する
- 12. PC から受信する
- 13. Unity の準備をする①

- 14. Unity の準備をする②
- 15. Unity でデータを受信する
- 16. 受信用スクリプトを書く
- 17. 受信用スクリプト概要①
- 18. 受信用スクリプト概要②
- 19. スケッチ側を書き込む
- 20. Unity の出力
- 21. Unity からデータを送信する
- 22. 送信用スクリプトを書く
- 23. 送信用スクリプト概要
- 24. スケッチとキー押し
- 25. 送信用スクリプトおまけ
- 26. 質問あります

1. シリアル通信とは • 電気通信において伝送路上を一度に1ビットず つ、逐次的にデータを送ることをいう。また、コン ピュータにおいては、バス上を一度に1ビットずつ、 逐次的にデータを送ることをいう。 ※Wikipedia より 0 1 0 0 1 デバイスB デバイスA あるデバイスから他のデバイス ヘデータを送る。 一本の電線に01の信号をのせ 一定時間間隔で送信する。



マイコン同士でシリアル通信をする場合は各デバイスに通信速度(01の信号を送る時間間隔)を設定してやる必要がある。



3. 通信速度の設定がずれると・・・

マイコンBの通信速度の設定がマイコンAの2倍
 だった場合



4. どうやって PC とつなぐの?

 すでにつないでいるんです。
 <u>なんと、シリアル通信を使えば、そのまま PCと</u> データのやり取りができる!



5.シリアル-USB 変換 IC

• Arduino 上での配置



6.PC にデータを送信してみよう

ボタンを押すとLED_Lが点灯し、シリアル通信で3
 種類のデータをPCへ送信するプログラム

Serial.begin(9600); →マイコン側の通信速度を設定 9600bps(ビット/秒)にする。 Serial.print(66); → 文字データ 66 を送信する。 ("66") でも OK 。 Serial.write(66); →バイナリデータ(01のデータ) 「66」を送信する。 Serial.write("B"); →文字「B」をバイナリデータで 送信する。

Serial.print(″¥n″); →改行する。

スケッチ SirialCom 参照

SirialCom Arduino 1.6.5		×	
ファイル 編集 スケッチ ツール ヘルプ			
		<mark>ب</mark> ور	
SirialCom §			
<pre>void setup() { // put your setup code here, to run once: pinMode(13,0UTPUT); //13pinを出力に設定。 pinMode(11,INPUT); //11pinを入力に設定。 Serial.begin(9600); }</pre>			^
<pre>void loop() { // put your main code here, to run repeatedly: if(digitalRead(11) == LOW){</pre>			
digitalWrite(13,LOW); //13pinの出力がLow(0V)になる。	*		
3			~

7.ASCIIコード

• 文字データBはバイナリデータで 66 となる。

文 字	10 進	16 進	文 字	10 進	16 進																		
NUL	0	00	DLE	16	10	SP	32	20	0	48	30	@	64	40	Ρ	80	50	•	96	60	р	112	70
SOH	1	01	DC1	17	11	!	33	21	1	49	31	Α	65	41	Q	81	51	а	97	61	q	113	71
STX	2	02	DC2	18	12	"	34	22	2	50	32	В	66	42	R	82	52	b	98	62	r	114	72
ETX	3	03	DC3	19	13	#	35	23	3	51	33	С	67	43	S	83	53	С	99	63	S	115	73
EOT	4	04	DC4	20	14	\$	36	24	4	52	34	D	68	44	Т	84	54	d	100	64	t	116	74
ENQ	5	05	NAK	21	15	%	37	25	5	53	35	Е	69	45	U	85	55	е	101	65	u	117	75
ACK	6	06	SYN	22	16	&	38	26	6	54	36	F	70	46	V	86	56	f	102	66	V	118	76
BEL	7	07	ETB	23	17	1	39	27	7	55	37	G	71	47	W	87	57	g	103	67	W	119	77
BS	8	80	CAN	24	18	(40	28	8	56	38	Н	72	48	Х	88	58	h	104	68	X	120	78
ΗT	9	09	EM	25	19)	41	29	9	57	39	Ι	73	49	Υ	89	59	i	105	69	У	121	79
LF*	10	0a	SUB	26	1a	*	42	2a	:	58	3a	J	74	4a	Ζ	90	5a	j	106	6a	Z	122	7a
VT	11	0b	ESC	27	1b	+	43	2b	;	59	3b	Κ	75	4b	[91	5b	k	107	6b	{	123	7b
FF*	12	0c	FS	28	1c	,	44	2c	<	60	3c	L	76	4c	\¥	92	5c	I	108	6c		124	7c
CR	13	0d	GS	29	1d	-	45	2d	=	61	3d	Μ	77	4d]	93	5d	m	109	6d	}	125	7d
SO	14	0e	RS	30	1e		46	2e	>	62	3e	Ν	78	4e	^	94	5e	n	110	<mark>6</mark> e	~	126	7e
SI	15	Of	US	31	1f	/	47	2f	?	63	3f	0	79	4f		95	5f	0	111	6f	DEL	127	7f

「IT 用語辞典 HP より」

8.シリアルモニタ

 Arduinoのデータ受信用ツール「シリアルモニタ」を 使用し、PCでデータを受信しよう。



9.スイッチ入力プログラム

プログラムをArduinoに書込み、スイッチを押す

プログラムを書き込む
SirialCom Arduino 1.6.5 -
SirialCom
<pre>void setup() { // put your setup code here, to run once: pinMode(13,0UTPUT); //13pinを出力に設定。 pinMode(11,INPUT); //11pinを入力に設定。 Serial.begin(9600); }</pre>
<pre>void loop() { // put your main code here, to run repeatedly: if(digitalRead(11) == LOW){ //11pinの入力がLow(0V)の場合、 digitalWrite(13,HIGH); //13pinの出力がHigh(5V)になる。 Serial.print(66); //シリアルモニタ*66* Unity*54*×2</pre>
Serial.print("¥n"); //政行 Serial.write(66); //シリアルモニタ"B"Unity"66" Serial.print("¥n"); Serial.write("B"); //シリアルモニタ"B"Unity"66"
<pre>Serial.print("¥n"); }</pre>
else //それ以外(11pinの入力がHigh)の場合、 digitalWrite(13,LOW); //13pinの出力がLow(OV)になる。
1 V
マイコンボードに書き込んでいます
グローバル変数が 186バイト (9%) の 動的メモリを使用しており、ローカル変数こ ^ 1,862 バイトが残っています。最高は 2,048バイトです。
20 Arduino/Genuino Uno on COM3

プログラムの通りLED_L が点灯、 シリアル-USB 変換 IC がデータを送信する ため、LED_TX が点灯する。



スイッチを押す。

「第2回 7. スイッチ」参照

10.シリアルモニタの表示

ボタンを押すとシリアルモニタ上に「66(改行)
 B(改行)B(改行)」が繰り返し表示される。



11.PC から受信する

バイナリデータ「66」を受信するとLED_Lが光るプログラムを書き込む
 スケッチ SirialRead 参照

int InByte;

→ int (整数型)の変数 InByteを 宣言する。InByte に数値を入 れて使用できるようになる

Serial.available() > 0 →Serial.available() は受信したデ ータのバイト数が入る。0 以上 なら受信データが存在すると いうことになる。

InByte = Serial.read(); → Serial.read() で受信したデータ (バイナリ)を読み込む。読み 込み可能なデータの最初の 1 バイトが入る。これが InByte に数値として入る。

```
ΘĐ
                     SirialRead | Arduino 1.6.5
ファイル 編集 スケッチ ツール ヘルプ
  SirialRead
int InByte;
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
                       //13pinを出力に設定。
  Serial.begin(9600);
void loop() {
  delay(1);
  if (Serial.available() > 0) { // 受信したデータが存在する
    InByte = Serial.read(); // 受信データを読み込む
  }
  if (InByte == 66)
   digitalWrite(13,HIGH) ;
  else
    digitalWrite(13,LOW) ;
}
```

12.PC から受信する シリアルモニタを起動する。



13.Unity の準備をする①

• Unity を起動する。

€			Unity Personal (64bit)) - Untitled - ShizuApp_SerialCom - PC, Mac & Linux Standalone <dx< th=""><th>11> - • ×</th></dx<>	11> - • ×
File Edi	it Assets GameObject	Component Wind	low Help		
۳	Undo	Ctrl+Z			🛆 Account 🔹 Layers 👻 Layout 💌
i≡ Hi	Redo	Ctrl+Y	Came 🕅 Animator	-= O Inspector	
Crea	Cut	Ctrl+X	•	Maximize on Play Mute audio Stats Gizmos ▼	
Dir	Сору	Ctrl+C			
	Paste	Ctrl+V			
	Duplicate	Ctrl+D			
	Delete	Shift+Del			
	Frame Selected	F			
	Lock View to Selected	Shift+F			
	Find	Ctrl+F			
	Select All	Ctrl+A			
	Preferences				
	Modules				
	Play	Ctrl+P			
	Pause	Ctrl+Shift+P			
	Step	Ctrl+Alt+P			
	Sign in				
	Sign out				
	Selection	•			
🕮 Pr	Project Settings	۱.	Input		
Crea	Network Emulation	•	Tags and Layers		
	Graphics Emulation	+	Audio		
(Snap Settings		Player	→Project Setting	
Q AI	l Scripts		Physics		
ass 🖹	ets		Physics 2D	→Player	
			Quality	を選択	
			Graphics	で歴れ。	
			Network		
			Script Execution Order		

14.Unityの準備をする②

• Unity を起動する。

<		Unity Personal (64bit) - Untitled - ShizuApp_SerialCom	- PC, Ma	c & Linux Standalone <dx11></dx11>	> ×
File Edit Assets GameObject C	Component \	Vindow Help			
	enter 🛛 🕸 Loc				🛆 Account 🔹 Layers 🔹 Layout 👻
[•] ≔ Hierarchy •≡	# Scene	C Game 🕺 Animator	*≡	Inspector	<u> -</u> =
Create * Q*All	Free Aspect	* Maximize on Play Mute audio Stats	Gizmos 🔻	Default Cursor	None
Main Camera					(Texture2D)
Directional Light					
				Cursor Hotspot	X 0 Y 0
				● ₹ □	* 43 0 0 5 0
				Settings for PC, Mac & Linux Standal	lone
				Resolution and Presentation	
				Icon	
				Splash Image	
			_	Other Settings	
				Rendering	
				Rendering Path*	Forward +
				Color Space*	Gamma +
				Auto Graphics API for Windows	
		PC Setting		Auto Graphics API for Linux	
				Static Batching	
		→Ontimization		GPU Skipping*	
		optimzation		Stereoscopic rendering*	
		→Ani Comnatihilitiv I evel		Virtual Reality Supported	
Project 🗄 Console				Configuration	
		「INEI Z.U Subset」を		Scripting Backend	Mono2x +
Q All Materials		「 ハгт 。 。 , ノー赤 臿		Disable HW Statistics	
Q All Models		Ⅰ.INE I Z.U]I _		Scripting Define Symbols	
Q All Prefabs					
				Optimization	
Assets				Api Compatibility Level	.NET 2.0 +
				Prebake Collision Meshes	✓ .NET 2.0
Sv	/stem	10 Ports を正義して		Preload Shaders	.NET 2.0 Subset
		- トニッジは田ゴキリーナッフ		Preloaded Assets	Minud
Se	erialP	ort ソフムか使用可能になる		Optimize Mesh Data*	Mixea +
				opunize Plesit Data	
				* Shared setting between multiple platf	forms.
ProjectSettings	s.asset				T

15.Unity でデータを受信する

• カメラ(とりあえず)にスクリプトを追加する。

◀ Unity Personal (64bit) - Untitled - ShizuApp_SerialCom - PC, Mac & Linux Standalone <dx11></dx11>	- 🗆 ×
File Edit Assets GameObject Component Window Help	
Image: Solution of the second sec	rs 🔻 Layout 👻
The Hierarchy -= # Scene Game Animator -= O Inspector	<u> -</u> =
Create * (QrAll) Free Aspect * Maximize on Play Mute audio Stats Gizmos * Main Camera	Static 👻
Main Camera Tag MainCamera + Layer Default	÷
Position X 0 Y 1	Z -10
Rotation X 0 Y 0	Z 0
Scale X 1 Y 1	Z 1
V∰ M Camera Clear Flags Skybox	₽ ,
Background	
Culling Mask	\$
Projection Name	+
Field of View SerialCom	60
Language C Sharp +	
Viewport Rect	
Depth Rendering Rath	
Target Texture	0
Occlusion Culling	
HDR	
▼	2 0,
Project Console	
Q All Materials	
Q All Models This folder is empty	
Q All Scripts	
Main Gamera	
→Add Component	
→New Script	
「」を追加する。	
$= -\frac{1}{7} \frac{1}{1} \int \frac{1}{10} \frac{1}{10$	
※ スケッチと名前がかぶっていろが別のプロ/	ゲラム 🗖

16.受信用スクリプトを書く ・データ受信プログラム。書いた後はビルドする。

```
SirialCom.cs 参照
                  SerialCom.cs + X
ty - Scripting API:

 Opdate()

ShizuApp SerialCom.CSharp

    SerialCom

using UnityEngine;
 using System.Collections;
                            //シリアルポートを制御するクラス
 using System.IO.Ports;
public class SerialCom : MonoBehaviour {
     SerialPort Port1 = new SerialPort("COM3", 9600); //通信ポートと通信速度の設定
     int data:
     // Use this for initialization
     void Start ()
     {
        if (Port1.IsOpen)
                            ノノシリアル ポートが開いている場合は true。それ以外の場合は false
        {
                         //ポート接続を閉じ、IsOpen プロパティを false に設定する。
            Port1.Close();
        else {
           Port1.Open();
                              //新しいシリアル ポート接続を開きます。
           Port1.ReadTimeout = 1000;
                                      //読み取り操作が完了していないときに、1000ms後にタイムアウトになります。
     // Update is called once per frame
     void Update () {
        data = 0;
        data = Port1.ReadByte();
                                   //SerialPort の入力バッファーから、同期で 1 バイトを読み取り、dataに入れます。
        Debug.Log(data);
```

tput

17.受信用スクリプト概要①

using System.IO.Ports; →シリアルポートを制御するクラスを追加する。 Setting を「.NET 2.0」にしたことで使用可能になる。

SerialPort Port1 = new SerialPort("COM3", 9600);

→ Port 番号の設定と通信速度を設定する。 Port 番号はスケッチのツールで設定した ものと同じ、または、デバイスマネージャー から確認できる。

第一回で確認したツールから設定した Port 番号 または、PC のデバイスマネージャーを開くと、 ポート(COM と LPT)内で確認できる

•	sketch_oct18a Arduino 1.6.5	×		
ファイル 編集 スケッチ	ツール ヘルプ			
	自動整形	Ctrl+T		
	スケッチをアーカイブする			
sketch_oct18a	エンコーディングを少し			
void setup() {	シリアルモニタ	Ctrl+Shift+M	^	
// put your setur pinMode(13,00TPU)	ボード: "Arduino/Genuino Uno"	,		
}	ポート: "COM3 (Arduino/Genuino Uno)"	,		Serial ports
void loop() {	書は法案・"AVRISP mkII"		~	COM3 (Arduino/Genuino Uno)
// put your main	ブートロータを含くたか			
<pre>digitalwrite(18, n_ }</pre>				
				1



18.受信用スクリプト概要②

```
if (Port1.IsOpen)
      Port1.Close():
    else {
      Port1.Open();
      Port1.ReadTimeout = 1000;
 →シリアルポートを開くプログラム。
   Timeout がないと、エラーが発生するので注意。設定した数値 ms後
   読み取り操作が完了していないとタイムアウトになる。
data = Port1.ReadByte();
 →シリアルポートから1バイト読み込む。
  読み込んだ値は data に書き込まれる。
Debug.Log(data);
 →コンソールに data の数値を出力します。
```

19.スケッチ側を書き込む

スケッチ SirialCom より" Serial.write(66)" 以外の
 送信をコメントアウトした後、書込みを行う。

スケッチ SirialCom2 参照	Unity Personal (64bit) - Main.unity - Shisarapp_scialCom - PC, Mac & Linux Standalone <dx11> Help</dx11>
SirialCom2 Arduino 1.6.5 - ロ × アイル 編集 スケッチ ツール ヘルプ	C Game S Animator C Game S Animator C Game S Animator C Inspector SerialCom
○○■□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	SerialCom using UnityEngine; using System.Colle using System.IO.P
<pre>void setup() { // put your setup code here, to run once: pinMode(13,0UTPUT); //13pinを出力に設定。 pinMode(11,INPUT); //11pinを入力に設定。 Serial begin(9600);</pre>	書込み後、Unity でゲームを実行し、 ブレッドボードのスイッチを押す。
<pre> void loop() () // put your main code here, to run repeatedly: if(digitalRead(11) == LOW) { //11pinの入力がLow(OV)の場合、 digitalWrite(13,HIGH) ; //13pinの出力がHigh(5V)になる。 // Serial.print("66"); //シリアルモニタ"66" Unity"54"×2 // Serial.print("¥n"); //改行 Serial.print("¥n"); //ひリアルモニタ"B" Unity"66" // Serial.print("¥n"); } else //それ以外(11pinの入力がHigh)の場合、 digitalWrite(13,LOW) ; //13pinの出力がLow(OV)になる。 } </pre>	Aryflar 時のみ 押すこと

20 .Unity の出力



21.Unity からデータを送信する

・カメラにスクリプトを追加する。

€			Unity	Personal (64bit) - M	lain.unity - Sl	hizuApp_S	erialCom - P	C, Mac & Linux Standa	alone* <dx11< th=""><th>></th><th></th><th>×</th></dx11<>	>		×
File Ed	dit Asset	GameObject	t Compon	ent Window	Help								
8	0 G	10 II I	■ Center	😂 Local							Layers	▼ Layout	•
i≡ Hiera	rchy		-= #	Scene	C Game	88 Anima	tor		0 Inspector				â -=
Create *	QTAII		Fi	ee Aspect	*	Maximize on Play	Mute audio	Stats Gizmos 🔻	Main Camera				Static 👻
Main C	Camera								Tag MainCamera	¢ [ayer Default		\$
Directi	ional Light								▼ 🙏 Transform				🛐 🌣,
									Position	× 0	Y 1	Z -10	
									Rotation	X 0	Y O	Z 0	
									Scale	X [1	Y 1	Z 1	
									Clear Flags	Skyboy			A 10
													9
									Serial	Com II	は使われ	ないたい	め 🗖
									Pro	エー・ハク	カンナ		
						0			Fiel	アエツン	2719 d)	FI
	Mair	ı Came	era II	新しし	いスク	リプト			Clipping Planes	Near 0,7			
	ГО		, u ,	· `白 뉴བ	+7	/ / /				Far _000			
	1 Se	rialRea	ad]を	[追加	9 00)			Viewport Rect	X	Y 0		
	\times 7	しケッチ	- レタ	前がっ	かごー	っていス	、が			·			
				יי גי ויק - י	J * /] * .)/J*		Depth	-1			
	ን	刂のフロ	コクラ	ラム				a .=	Rendering Path	Use Player	Settings der Texture)		
c			• •						Occlusion Culling		ider rexture)		
▼☆Fav	orites	Assets							HDR				
	II Material:								🔻 🖵 🗹 GUL Layer				💽 🔅,
l õÃ	Il Prefabs		C	C II					🛛 📄 🗹 Fiare Layer				💽 🗘,
Q A	Il Scripts		C #	C#					V 🖉 🖸 Audio Listener				🔯 🌣,
	ate	Main	SerialCom	SerialRead					🔻 🧃 🗔 Serial Com (Scrij	pt)			💽 🌣,
ASS	sets								Script	© SerialCo	om		0
									🔻 健 🗹 Serial Read (Scri	ipt)			🔄 🌣,
									Script	@ SerialRe	ad		
										Add Comr	oonent		

22.送信用スクリプトを書く ・データ送信プログラム。書いた後はビルドする。

```
SirialRead.cs 参照
Unity - Scripting API:
                    SerialRead.cs
                                                                                                   - ©<sub>a</sub> Ur
C# Assembly-CSharp-vs
                                                  🔩 SerialRead
  using UnityEngine;
    using System.Collections;
    using System.IO.Ports;
                              //シリアルポートを制御するクラス
  public class SerialRead : MonoBehaviour {
       SerialPort Port1 = new SerialPort("COM3", 9600); //通信ポートと通信速度の設定
       // Use this for initialization
       void Start ()
                             //シリアル ポートが開いている場合は true。それ以外の場合は false
           if (Port1.IsOpen)
           {
                             //ポート接続を閉じ、IsOpen プロパティを false に設定する。
              Port1.Close();
           else {
              Port1.Open();
                               //新しいシリアル ポート接続を開きます。
                                        //読み取り操作が完了していないときに、1000ms後にタイムアウトになります。
              Port1.ReadTimeout = 1000:
           3
       }
       // Update is called once per frame
       void Update () {
           if (Input.GetKey(KeyCode.N))
                                        //Nキーを押すと { 以下を実行する
              Debug.Log("N");
              byte[] byte1 = new byte[1];
              byte1[0] = 66;
                                        //byte1を送信
              Port1.Write(byte1, 0, 1);
```

23.送信用スクリプト概要

If (Input.GetKey (KeyCode.N)) { } →キーボードのNキーが押されると{ }内のコードが実行される。

Debug.Log (″N ″); →コンソールにNが出力します。

Byte [] byte1 = new byte [1];

Byte 1[0] = 66;

Port1.Write (byte1, 0, 1); →「byte1」の配列の「0」列目から「1」バイトを送信

配列・・・一つの変数に複数の数値を定義できる。 ex. Byte byte1[3]={66,100,8} ※3つの数値を定義したバイト型の配列の例 0列目に66(byte1[0]=66)、1列目に100(byte1[1]=100)、 2列目に8(byte1[2]=8)が格納されている。

24.スケッチとキー押し

11. で使用したスケッチを Arduino に書き込み、
 キーボードのNを押す。

スケッチ SirialRead 参照

		TVUIDO
SirialRead Arduino 1.6.5		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
ファイル 編集 スケッチ ツール ヘルプ	Maximize on Play Mute audio	GHJKI
SirialRead	÷.	v z t, ta
int InByte;		
void setup() { // put your setup code here, to run once: pinMode(13,0UTPUT); //13pinを出力に設定。 Serial.begin(9600);	書込み後、Unity て キーボードの N を押 ⁻	でゲームを実行し、 すとLED が点灯する
<pre>void loop() { delay(1); if (Serial.available() > 0) { // 受信したデータが存在する InByte = Serial.read(); // 受信データを読み込む } if (InByte == 66) digitalWrite(13,HIGH); else digitalWrite(13,LOW); }</pre>		

25.送信用スクリプトおまけ 以下の一文だけでも66の送信ができる。 キーを押すと一瞬だけLEDLが点灯する。 (続いてCRとLFが送信されるため)

SirialRead2.cs 参照 // Update is called once per frame void Update () { Ė if (Input.GetKey(KeyCode.N)) Debug.Log("N"); Port1.WriteLine("B");

Port1.WriteLine("B"); →文字列「B」を送信する。 ① 66 アスキーコード「B」 2 13 アスキーコード「CR(復帰)」 ③ 10 アスキーコード「LF(改行)」 の順に数値が送信される。

26. 質問あります

- •「.NET 2.0」は「.NET 2.0 Subset」と何が違うの?
- Unity 側の読み込みスピードを上げたい。
 どうすればよいのか?
- スピードが追い付いてない場合、Unity 側で調整 できるの?









ご清聴ありがとうございました。