実験レポート 30 for JA1YZT ／茨城県立太田第一高校ｱﾏﾁｭｱ無線ｸﾗﾌﾞ(物理部)　　2026.03.10

「RaspberryPiでCW解読・CWで歌おう！」

※ CW ＝電信＝ﾓｰﾙｽ信号

０．概　要：

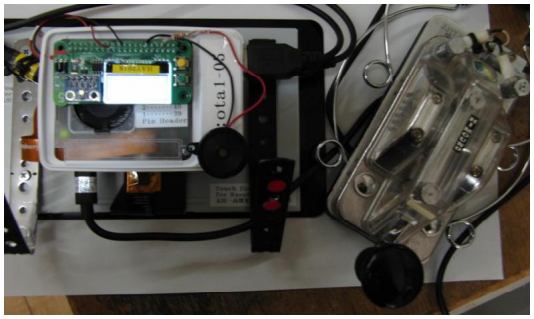
Raspberry Pi 5 で CW(ｽﾄﾚｰﾄｷｰ入力によるﾓｰﾙｽ(電信)信号)解読、
CW に音階音符をつけて「ABC の歌」を歌っちゃおう！
(Soft：python3、／ Hard：Pi5 の GPIO23=KEYin、GPIO16=圧電ﾌﾞｻﾞｰ、他 LED 等)

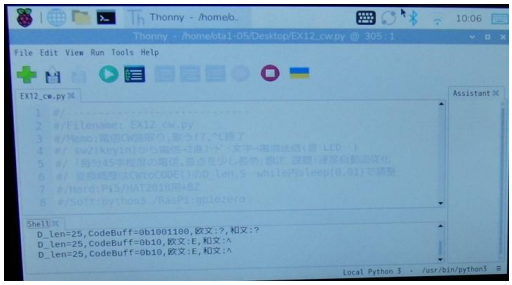
Ⅰ．はじめに：

2025.12 本校探究推進部が千葉工大：千秋博紀先生を招聘して「ﾗｽﾞﾍﾞﾘｰﾊﾟｲの使い方」講座を開催、おいしそうな名前に惹かれて物理部が相乗りした。がっ！
Linux/Unix って何？、python って何?、気象観測ﾌﾟﾛｸﾞﾗﾑ log.py って何？‥
ｽｲｯﾁや LED「Ｌﾁｶ」、I2C による LCD ／気象ｾﾝｻｰ／加速度ｾﾝｻｰ制御、ADC、
ｻｰﾎﾞﾓｰﾀｰ‥と実験を重ねる中、「音がないのが寂しい」→圧電ﾌﾞｻﾞｰによる演奏、
「ｽｲｯﾁ 1 個でいろいろな命令を入力できない？」　→ﾓｰﾙｽ信号の自動解読、
「ﾓｰﾙｽ信号で歌えない？」‥物理部のわがままは続く･･･

Ⅱ．製作・実験：

（１）Hard:Pi5 の「 GPIO23 入力=KEYin、GPIO16 出力=圧電ﾌﾞｻﾞｰ」、その他 LED や PTT
（２）Soft:Python3 にて Thonny で製作したﾌﾟﾛｸﾞﾗﾑﾘｽﾄ　(以下 cw.py)
（３）音階表、ﾓｰﾙｽｺｰﾄﾞ表、等　(以下)





←一式　↑変換文字が下の Shell 画面に表示

Ⅲ．結果・課題：

（１）音響：gpiozero のﾌﾞｻﾞｰ命令では 1 ｵｸﾀｰﾌﾞしか出なかったので、別途関数製作
（２）CW 簡易解読：1 文字ずつ変換→「欧文・和文」表示。速度自動追従できればいいな‥
（３）CW 文連続出力 OK ！→ CW で歌ってしまえっ！　8bit ﾒﾓﾘで処理したが、もっと最新化が?

Ⅳ．参考資料：(敬称略／原典の公開ｿｰｽｺｰﾄﾞを物理部が勝手に解釈・再ﾌﾟﾛｸﾞﾗﾑ、原典の性能を保証できない)

①「RaspberryPi で学ぶ電子工作」金丸隆志　(技術評論社 2024)
②「温湿度・気圧を手軽に測ろう」小野寺康之 (ﾗｽﾞﾊﾟｲﾏｶﾞｼﾞﾝ 2018・10/日経 BP)
③「特集 2 主要電子ﾊﾟｰﾂ入門」　(ﾗｽﾞﾊﾟｲﾏｶﾞｼﾞﾝ 2025・秋/日経 BP)
④「ｴﾚｷｼﾞｬｯｸ 20 号」PIC 記事 光永法明　(CQ 出版)　→ﾌﾞｻﾞｰ演奏関連
⑤「PC-6001 によるﾓｰﾙｽ信号の解読実験ﾌﾟﾛｸﾞﾗﾑ」JI1NZL/青木昇　→電信解析
(CQ ﾊﾑﾗｼﾞｵ 1991.4/CQ 出版) **Basic** ＋ ﾈｯﾄ上､詠み人知らずの公開ｱﾙｺﾞﾘｽﾞﾑ
→ **C** 言語/PIC12F1822(+改)　(JA1YEF/茨城県立日立第一高校物理部 2013.3)
→ **Python**/RasPi5(+和,song)　(JA1YZT/茨城県立太田第一高校物理部 2026.3)

追記：〈ﾓｰﾙｽ信号事始め〉2012 年、1 年生の遠足で茨城県阿見町にある「予科練平和記念館」を訪れた。自分たちに近い年代の若者が訓練・特攻へ‥「敵艦見ﾕ」打電ののちﾊﾟｲﾛｯﾄ自らが電信連続音発出で突撃、電信音の尽きたときが死を迎えたとき･･･この残酷さに皆涙した。このときの「電信って何？」石碑に刻まれた「若鷲の歌」の楽譜から、「電信解析・再現、ﾒﾛﾃﾞｨｰ演奏」ﾌﾟﾛｼﾞｪｸﾄが当時の物理部で始動、PIC ｵﾙｺﾞｰﾙとなった。その後、2014 茨城総文祭無線部会の工作会、SSH の人工衛星ﾓﾄﾞｷや ARDF 送信機のﾋﾞｰｺﾝ制御に発展した。天気晴朗なれど波高き現在、平和を祈念しつつ。

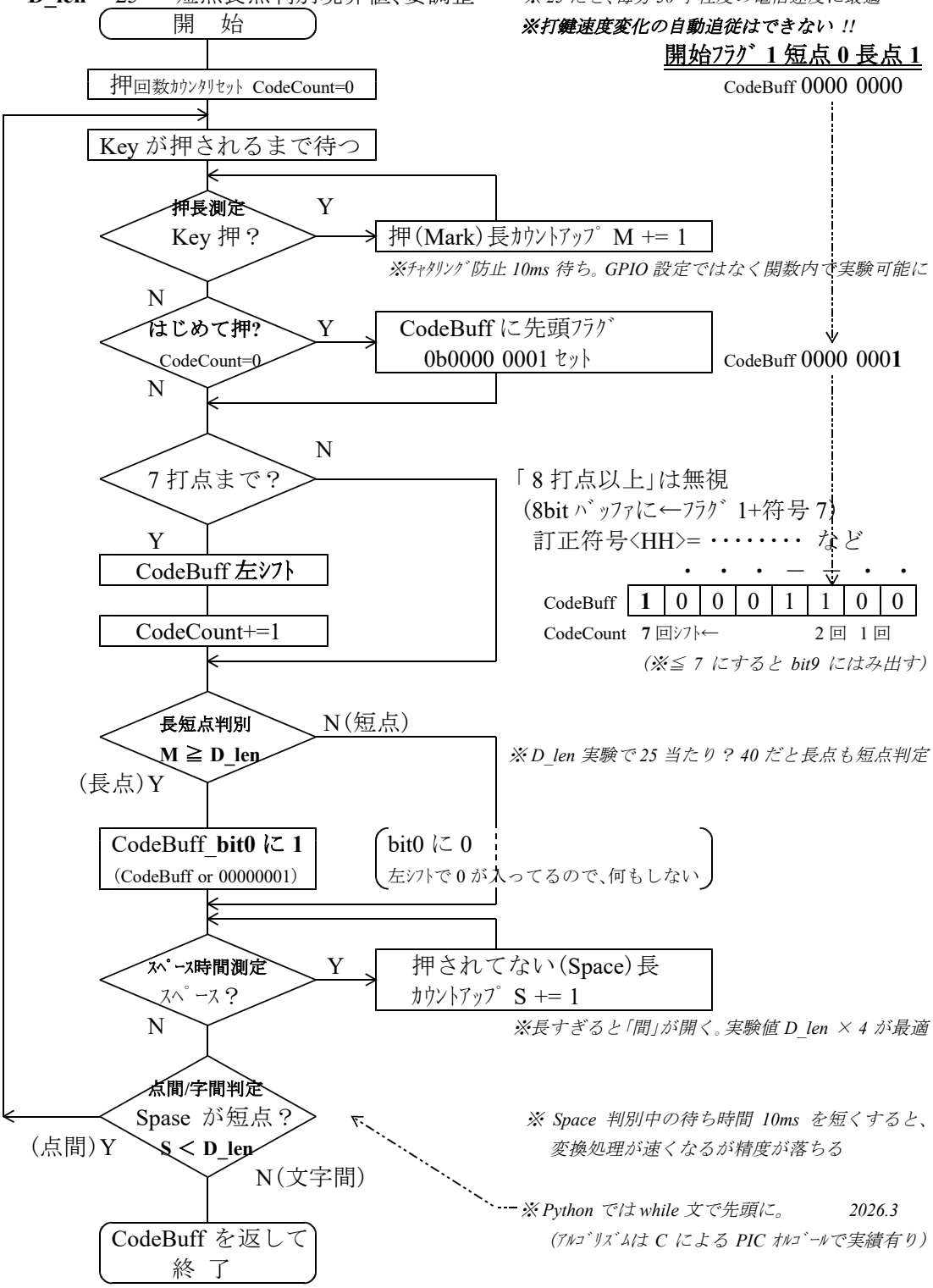
2026.3　for JA1YZT　　by JJ1TJK ／関山 大志

＜モールス信号１字受信＞　　a= CWtoCODE()関数　　※ は実験ｺﾒﾝﾄ

（電鍵を押す Mark ／離す Space,　短点 dot ／長点 dash,　dash=3dot,打点間 1dot,文字間 3dot,単語間 7dot）

D_len = 25･･･短点長点判別境界値、要調整　　*※25 だと、毎分 50 字程度の電信速度に最適*

※打鍵速度変化の自動追従はできない !!



☆別関数にて以下のﾃｰﾌﾞﾙ参照で「ｺｰﾄﾞ→文字」変換

CW ｺｰﾄﾞ_ﾃｰﾌﾞﾙ CWabc_CODE=[0b0000 0101, 0b0001 1000, ･･･] 開始ﾌﾗｸﾞ 1 短点 0 長点 1

文字_ﾃｰﾌﾞﾙ　　abc_table　　= ['A', 'B', ･･･　] 順番が上記ﾃｰﾌﾞﾙに対応

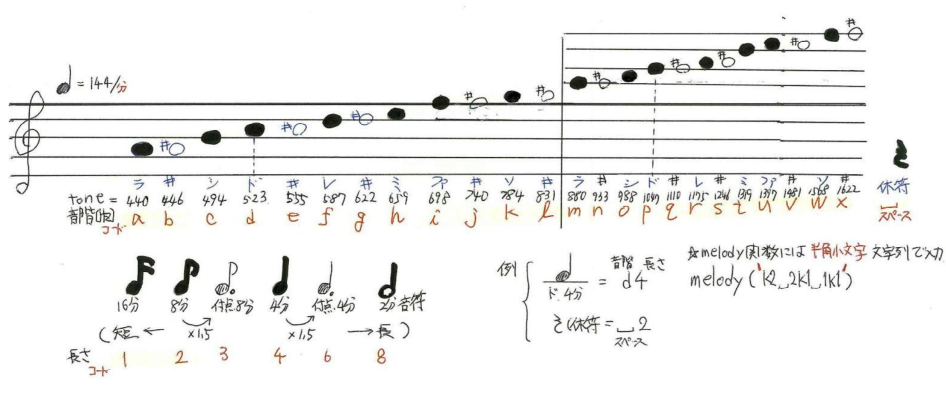
電信（CW）ｺｰﾄﾞ

☆無線通話表 — 電信Continuous Wave (Morse) — Phonetic Code — 2026/3/5 — ﾓｰﾙｽ自動解析Code

和	②和文通話法	No.	(和文合調法)	和	④モールス符号③	欧	(欧文合調法)	欧	①欧文通話法		<HAM>	ﾌﾗｸﾞ 1短0長1	16進	ASCII10	ASCII16	頻度順Δt
イ	いろはの い	1	伊藤	イ	・－	A	亜鈴(ｱﾚｰ)	A	Alfa	アルファ	ｱﾙﾌｧ	00000101	05	65	41	5
ロ	ローマの ろ	2	路上歩行	ロ	・－・－							00010101	15			
ハ	葉書の は	3	ハーモニカ	ハ	－・・・	B	棒倒し	B	Bravo	ブラヴォウ	ﾌﾞﾗﾎﾞｰ	00011000	18	66	42	9
ニ	日本の に	4	入費超過	ニ	－・－・	C	チャートルーム	C	Charlie	チャーリ	ﾁｬｰﾘｰ	00011010	1A	67	43	11
ホ	保険の ほ	5	宝石	ホ	－・・	D	道徳	D	Delta	デルタ	ﾃﾞﾙﾀ	00001100	0C	68	44	7
ヘ	平和の へ	6	屁	ヘ	・	E	エ	E	Echo	エコウ	ｴｺｰ	00000010	02	69	45	1
ト	東京の と	7	特等席	ト	・・－・・							00100100	24			
チ	ちどりの ち	8	知己多し	チ	・・－・	F	古道具	F	Foxtrot	フォクストロット	ﾌｫｯｸｽﾄﾛｯﾄ	00010010	12	70	46	9
リ	リンゴの り	9	流行す	リ	－－・	G	強情だ	G	Golf	ゴルフ	ｺﾞﾙﾌ	00001110	0E	71	47	9
ヌ	沼津の ぬ	10	塗り壁	ヌ	・・・・	H	腹から	H	Hotel	ホウテル	ﾎﾃﾙ	00010000	10	72	48	7
ル	るすいの る	11	ルーム上等だ	ル	－・－－・	KN	(割込禁「どうぞ」)(					00110110	36	40	28	(
ヲ	尾張の を(wo)	12	和尚焼香	ヲ	・－－－	J	自由行動	J	Juliett	ジュウリエット	ｼﾞｭﾘｴｯﾄ	00010111	17	74	4A	13
ワ	わらびの わ	13	ワークデー	ワ	－・－	K	警視庁	K	Kilo	キーロウ	ｷﾛ	00001101	0D	75	4B	9
カ	為替(ｶﾜｾ)の か	14	加盟する	カ	・－・・	L	流浪する	L	Lima	リーマ	ﾘﾏ	00010100	14	76	4C	9
ヨ	吉野の よ	15	ヨーヨー	ヨ	－－	M	メーデー	M	Mike	マイク	ﾏｲｸ	00000111	07	77	4D	7
タ	タバコの た	16	タームterm	タ	－・	N	農夫	N	November	[illegible]	ﾉｰﾍﾞﾝﾊﾞｰ	00000110	06	78	4E	5
レ	レンゲの れ	17	礼砲数	レ	－－－	O	応急法	O	Oscar	オスカ	ｵｽｶｰ	00001111	0F	79	4F	11
ソ	そろばんの そ	18	曹長候補	ソ	－－－・							00011110	1E			
ツ	つるかめの つ	19	都合通知	ツ	・－－・	P	プレーボーイ	P	Papa	パパ	ﾊﾟﾊﾟ	00010110	16	80	50	11
ネ	ねずみの ね	20	ねーねー行こー	ネ	－－・－	Q	救急至急	Q	Quebec	ケベック	ｹﾍﾞｯｸ	00011101	1D	81	51	13
ナ	名古屋の な	21	仲人	ナ	・－・	R	レコード	R	Romeo	ロウミオウ	ﾛﾒｵ	00001010	0A	82	52	7
ラ	ラジオの ら	22	ラヂオ	ラ	・・・	S	進め	S	Sierra	シエラ	ｼｱﾗｰ	00001000	08	83	53	5
ム	無線の む	23	ムー	ム	－	T	ティー(tea)	T	Tango	タンゴ	ﾀﾝｺﾞ	00000011	03	84	54	3
ウ	上野の う	24	疑う(ｳﾀｺﾞｰ)	ウ	・・－	U	疑う(ｳﾀｺﾞｰ)	U	Uniform	ユーニフォーム	ﾕﾆﾌｫｰﾑ	00001001	09	85	55	7
ヰ	ゐどの ゐ(wi)	25	行こー良かろー	ヰ	・－・・－				<慣用句(電信)>			00101001	29			
ノ	野原の の	26	乃木東郷	ノ	・・－－				HI	ｱﾊﾊ(笑い)		000010011	13			
オ	大阪の お	27	思う所	オ	・－・・・				ES/C	and/Yes		00101000	28			
ク	クラブの く	28	苦しそー	ク	・・・－	V	ビクトリー	V	Victor	ヴィクタ	ｳﾞｨｸﾀｰ	00010001	11	86	56	9
ヤ	大和(ﾔﾏﾄ)の や	29	野球場	ヤ	・－－	W	和洋風	W	Whiskey	ウィスキ	ｳｨｽｷｰ	00001011	0B	87	57	9
マ	マッチの ま	30	まー任そー	マ	－・・－	X	Xray(ｴｰｸｽﾚｲ)	X	X-ray	エクスレイ	ｴｯｸｽﾚｲ	00011001	19	88	58	11
ケ	景色の け	31	計器良好	ケ	－・－－	Y	養子孝行	Y	Yankee	ヤンキ	ﾔﾝｷｰ	00011011	1B	89	59	13
フ	富士山の ふ	32	風景無比	フ	－－・・	Z	ざーざー雨	Z	Zulu	ズール	ｽﾞﾙ	00011100	1C	90	5A	11
コ	子供の こ	33	皇統継承	コ	－－－－			<Q符号>	QRA/QTH	局名/位置(運用場所)		00011111	1F			
エ	英語の え	34	栄華十数世	エ	－・－－－				QRH	周波数変動		00110111	37			
テ	手紙の て	35	手数な修正	テ	・－・－－				QRL?	そちら通信中?(使用中?)		00101011	2B			
ア	朝日の あ	36	あー言うとこう言う	ア	－－・－－				QRM/QRN/QSB	混信/空電/フェージング		00111011	3B			
サ	桜の さ	37	サーモ表示計	サ	－・－・－				QRP/QRO	電力減少/増加		00110101	35			
キ	切手の き	38	聞ーて報告	キ	－・－・・				QRS	速度遅く送信して		00110100	34			
ユ	弓矢の ゆ	39	遊撃優秀	ユ	－・・－－				QRU/NIL	送信するもの無し		00110011	33			
メ	明治の め	40	姪からの状	メ	－・・・－	BT	(欧文本文)=		QRV/QRT	用意良(運用中)/閉局		00110001	31	61	3D	=
ミ	三笠の み	41	身のせーと言う	ミ	・・－・－				QRZ?	誰がこちらを呼んでいますか		00100101	25			
シ	新聞の し	42	少々スローだ	シ	－－・－・				QSA(ﾒﾘｯﾄ)/QRK	感度5/明瞭度5		00111010	3A			
ヱ	かぎのある ゑ(ゑwe)	43	恵方東北	ヱ	・－－・・				QSL	(確認/交信証)		00101100	2C			
ヒ	飛行機の ひ	44	表彰した例	ヒ	－－・・－				QSP	中継(伝える)		00111001	39			
モ	もみじの も	45	毛布と毛布	モ	－・・－・	/	(斜線・移動地)		QSY	周波数変更(移動して)		00110010	32	47	2F	
セ	世界の せ	46	世評傾聴す	セ	・－－－・				EX/VVV	試験中/本日は晴天なり		00101110	2E			
ス	すずめの す	47	数量表示計	ス	－－－・－				R/K/BK	了解/どうぞ/ブレーク(割込)		00111101	3D			
ン	おしまいの ん	48	んめーんめーナ	ン	・－・－・	AR	(送信[文]終了)		SOS(電信のみ)	メーデー、遭難		00101010	2A	13	0D	CR
゛	濁点(だくてん)	49	ダダ(濁点)	゛	・・	I	石	I	India	インディア	ｲﾝﾃﾞｨｱ	00000100	04	73	49	3
゜	半濁点	50	ポスター標語	゜	・・－－・				<慣用句(電信・通話)>			00100110	26			
ー	長音	51	貯蔵倉庫空(ｸｰ)	ー	・－－・－				CQ	各局		00101101	2D			
⌐	段落	52	次行の行から	⌐	・－・－・・				FB	すばらしい		01010100	54			
（	下向括弧	53	カーの上方を覆う	（	－・－－・－	)	(－・－－・		WX	天候		01101101	6D	41	29	
）	上向括弧	54	下方からページ(閉止)	）	・－・・－・				OM/YL/XYL	先輩/女性/妻		01010010	52			
、	区切点	55	切ろー切ろー切ろー	、	・－・－・－	.	(ピリオド)		73(88)	さようなら(女性へ)		01010101	55	46	2E	.
		56		?	・・－－・・	?	(疑問符)		RS(T)	了解度5信号強度9音調9		01001100	4C	63	3F	?
		57	(訂正・終了)	ラタ	・・・－・				OSO	非常通信		00100010	22			
		58	(和文本文)	ホレ	－・・－－－							01100111	67			
		59			・・・・・・・・	HH	(訂正「CORRECTION」)	⑤	海上無線	航空無線	<航空無線>	10000000	80	8	8	BS
					－－・・－－	,	(カンマ)	1000		タウザンド	TOU-SAND	01110011	73	44	2C	,
	和文数字				・－－・－・	@		100		ハンドレッド	HUN-dred	01011010	5A	64	40	@
.	小数点				・・・－・－	VA	(全電文終了)	.	DECIMAL	デシマル	DAY-SEE-MAL	01000101	45			VA
1	数字の ひと	60		1	・－－－－	1	A(略数字)	1	UNA-One	ワン	WUN	00101111	2F	49	31	
2	数字の に	61		2	・・－－－	2	U	2	BISSO-Two	ツー	TOO	00100111	27	50	32	
3	数字の さん	62	ミツキゆーこー	3	・・・－－	3	V	3	TERRA-Three	トリー	TREE	00100011	23	51	33	
4	数字の よん	63	よろこびりー	4	・・・・－	4	H	4	KARTE-Four	フォア	Fow-er	00100001	21	52	34	
5	数字の ご	64	ごもくめし	5	・・・・・	5		5	PANTA-Five	ファイフ	FIFE	00100000	20	53	35	
6	数字の ろく	65	ローソクたて	6	－・・・・	6		6	SOXI-Six	シクス	SIX	00110000	30	54	36	
7	数字の なな	66	なーむーななつ	7	－－・・・	7	B	7	SETTE-Seven	セヴン	SEV-en	00111000	38	55	37	
8	数字の はち	67	ヤーヤーヨーきた	8	－－－・・	8	D	8	OKTO-Eight	エイト	AIT	00111100	3C	56	38	
9	数字の きゅう	68	空中航空機	9	－－－－・	9	N	9	NOVE-Nine	ナイナ	NIN-er	00111110	3E	57	39	
0	数字の まる	69		0	－－－－－	0	T	0	NADA-Zero	ゼロウ	ZE-RO	00111111	3F	48	30	

音階ｺｰﾄﾞ（melody 関数用）

sound.py melody( )の入力値



```python
#/-----------------------------
#/Filename: EX12_cw.py
#/Memo:電信CW読取り,歌う!?,^C終了
#/ sw2(keyin)から電信→2進ｺｰﾄﾞ・文字→電信送信(音・LED・・)
#/「毎分45字程度の電信,長点を少し長め」想定。課題:速度自動追従化
#/ 変換精度はCWtoCODE()のD_len,S・・while内sleep(0.01)で調整
#/Hard:Pi5/HAT2018用+BZ
#/Soft:python3 /RasPi:gpiozero
#/Writer:by JJ1TJK for JA1YZT 2026/03/09
#/-----------------------------
#/**************************/
#/ declaration(宣言)
#/**************************/
### import文
import gpiozero #入出力(RasPi)ﾗｲﾌﾞﾗﾘ使用
import time             #時間ﾗｲﾌﾞﾗﾘ使用/LCD,BME280

### global variable(変数定義)/configuration(環境設定)& pin assign
#GPIO_SW1 = 22  #ｽｲｯﾁ1(緑)をGPIO22(負論理)
GPIO_SW2 = 23   #ｽｲｯﾁ2(黄)をGPIO23(負論理),CW_keyin
D_len=25#CW受信_dot(短点),dash(長点)判別境界値
CWtone=700#CW送信音周波数[Hz]
dot_len=0.06#CW送信速度(短点時間60ms)

GPIO_LED1 = 5   #LED1(青)をGPIO05[HAT2018(青)=5,ﾏﾙﾂHAT_R(赤)=17]
GPIO_LED2 = 6   #LED2(白)をGPIO06,CWkey_out
GPIO_BZ = 16    #BuzzerをGPIO16に [HAT2018改=16,ﾏﾙﾂHAT=23]
SPEED=0.05              #Buzzer_melody関数の曲の速さ50ms
#sw1 = gpiozero.DigitalInputDevice(GPIO_SW1, pull_up=True, bounce_time=0.05)#Mark長,Space長ｶｳﾝﾀｰ(0～225))
key_in = gpiozero.DigitalInputDevice(GPIO_SW2, pull_up=True)#sw2を電信入力に
    #GPIO23,ﾌﾟﾙｱｯﾌﾟ:押す(負論理)=Low(0V)→有効(sw.value=1)

led1 = gpiozero.DigitalOutputDevice( GPIO_LED1 )#GPIO05を出力､名称led1に
key_out= gpiozero.DigitalOutputDevice(GPIO_LED2)#GPIO06を出力､名称key_outに
bz = gpiozero.DigitalOutputDevice( GPIO_BZ )#GPIO16を出力、名称bz(ﾌﾞｻﾞｰ)に

#/**************************/
#/ functions(関数定義)
#/**************************/
###Buzzer_sound /gpiozero,time----------
def beep(f, l): #指定音(周波数f=frequency[Hz],音長l=length[s])
    for num in range(int(f*l)): #f回/1s→fl回/l秒→整数に
        bz.value=1                      #H
        time.sleep(1/(2*f))             #f[Hz]の半周期待つ
        bz.value=0                      #L
        time.sleep(1/(2*f))             #ここまで1周期

    #演奏関数 melody('a4')音程･長さ
    #音階：a=ﾗ440,b=#446,c=ｼ494,d=ﾄﾞ523,e=#555,f=ﾚ587,g=#622,h=ﾐ659,i=ﾌｧ698,j=#740,k=ｿ784,l=#831,
    #[Hz] m=ﾗ880,n=#933,o=ｼ988,p=ﾄﾞ1047,q=#1110,r=ﾚ1175,s=#1246,t=ﾐ1319,u=ﾌｧ1397,v=#1481,w=ｿ1568,x=#1622,
    #長さ：1～9 / 1=16分音符♬,2=8分音符♪,(3=付点8分),4=4分音符♩,(6=付点4分),8=2分,  「半角ｽﾍﾟｰｽ=休符」
    #速さ：SPEED=0.05(標準)  大→遅/無指定→標準 (楽譜表記：4分音符「♩=144」/分 → SPEED= 60s/(4*144) =0.1)
tone=[440,446,494,523,555,587,622,659,698,740,784,831,
      880,933,988,1047,1110,1175,1246,1319,1397,1481,1568,1622]#音階ﾃｰﾌﾞﾙ
SPEED = 0.05    #ﾒﾛﾃﾞｨｰ速度設定

def melody(s):  #演奏関数 melody('a4')音程･長さ
    global SPEED
    c=len(s)    #引き渡された文字列の長さ(1～)をｶｳﾝﾄ
    i=0
    for num in range(int(c*1/2)):       #文字数(1～)の半分回(整数)繰返し
        p=s[i]                          #文字列i番目(音程)
        l=s[i+1]                        #次の数字(〇分音符)
        #print(f'p={p},l={l}')#読込変数ﾁｪｯｸ用
        if(ord(p)>=ord('a') and ord(p)<=ord('x')) :     #'a'～'x'のとき｡ord()文字→ASCIIｺｰﾄﾞ
            beep(tone[ord(p)-ord('a')], (ord(l)-ord('0'))*SPEED)#音階ﾃｰﾌﾞﾙから周波数
        else:                           #'a'～'x'以外のとき=休符
            time.sleep((ord(l)-ord('0'))*SPEED) #休符
        i += 2
    SPEED = 0.05        #ﾒﾛﾃﾞｨｰ速度設定,指定無き場合は0.05


###CW(Morse)_decode function /gpiozero,time--------
    #key_in(ｽｲｯﾁ)から電信 →2進ｺｰﾄﾞ・文字 →音・LED・電信送信
    # KeyDown =L =Mark :  ・短点dot,  ー長点dash(=3dot)
    # KeyUp   =H =Space: 点間=1dot, 文字間=3dot, 単語間=7dot
    #'J'=・ーーー  =0001 0111(先頭ﾌﾗｸﾞ1,短点0,長点1)=0x17→CodeBuff
    #           先頭ﾌﾗｸﾞ,符号       →数             →CodeCount
    #欧文(利用頻度順)CWｺｰﾄﾞ_ﾃｰﾌﾞﾙ(49個／[0]～[48] @JA1YZT2026.3.5)
```

```
abc_table= ['E','T','I','A','N','S','M','U','R','D','H','J','Q',
           'W','K','G','V','F','L','B','O','P','X','C','Z','Y',
           '1','5','9','7','3','8','2','4','6','0','?','/',
           '<BT>','<AR>',',',';',')','(','@','<VA>',':','-','<HH>']
CWabc_CODE=[0x02,0x03,0x04,0x05,0x06,0x08,0x07,0x09,0x0A,0x0C,0x10,0x17,0x1D,
           0x0B,0x0D,0x0E,0x11,0x12,0x14,0x18,0x0F,0x16,0x19,0x1A,0x1C,0x1B,
           0x2F,0x20,0x3E,0x38,0x23,0x3C,0x27,0x21,0x30,0x3F,0x4C,0x32,
           0x31,0x2A,0x55,0x73,0x6D,0x36,0x5A,0x45,0x78,0x61,0x80]
    #和文CWｺｰﾄﾞ_ﾃｰﾌﾞﾙ(69個／[0]～[68])
wa_table = ['ｲ','ﾛ','ﾊ','ﾆ','ﾎ','ﾍ','ﾄ','ﾁ','ﾘ','ﾇ','ﾙ','ﾌ','ﾜ','ｶ',
           'ﾖ','ﾀ','ﾚ','ｿ','ﾂ','ﾈ','ﾅ','ﾗ','ﾑ','ｳ','ヰ','ﾉ','ｵ','ｸ','ﾔ','ﾏ',
           'ｹ','ﾌ','ｺ','ｴ','ﾃ','ｱ','ｻ','ｷ','ﾕ','ﾒ','ﾐ','ｼ','ヱ','ﾋ','ﾓ','ｾ','ｽ','ﾝ',
           '゛','゜','ｰ','｢','｣','(',')','?','<ﾗﾀ>','<ホレ>',
           '1','2','3','4','5','6','7','8','9','0','<HH>']
CWwa_CODE= [0x05,0x15,0x18,0x1A,0x0C,0x02,0x24,0x12,0x0E,0x10,0x36,0x17,0x0D,0x14,
           0x07,0x06,0x0F,0x1E,0x16,0x1D,0x0A,0x08,0x03,0x09,0x29,0x13,0x28,0x11,0x0B,0x19,
           0x1B,0x1C,0x1F,0x37,0x2B,0x3B,0x35,0x34,0x33,0x31,0x25,0x3A,0x2C,0x39,0x32,0x2E,0x3D,0x2A,
           0x04,0x26,0x2D,0x54,0x6D,0x52,0x55,0x4C,0x22,0x67,
           0x2F,0x27,0x23,0x21,0x20,0x30,0x38,0x3C,0x3E,0x3F,0x80]
    #CW受信→ｺｰﾄﾞ/文字 変換関連 (事前にkey_in入力GPIO設定が必要)
D_len=25          #短長点判別境界値25→大=遅(ｸﾞﾛｰﾊﾞﾙ変数)
def CWtoCODE(): #key_in→CW_CODE(電信入力1字をCWｺｰﾄﾞに変換)
    global D_len          #短長点判別境界値→大=遅(ｸﾞﾛｰﾊﾞﾙ変数)
    CodeBuff = 0          #変換ｺｰﾄﾞﾊﾞｯﾌｧ_ﾘｾｯﾄ
    CodeCount = 0         #Mark数ｶｳﾝﾀｰ_ﾘｾｯﾄ
    M=0                   #Mark長ｶｳﾝﾀｰ(0～225)
    S=0                   #Space長ｶｳﾝﾀｰ(0～225)

    while S < D_len:#Space=点間(1dot)の時,関数先頭(ここ)へ戻る
        while key_in.value != 1:          #keyが押されるまで待つ
            pass
        M=0      #Mark(keyが押された=L)時間測定
        while (key_in.value==1 and M<=250):#Markの間繰返し。長すぎる時抜ける
            time.sleep(0.01)    #10ms,ﾁｬﾀﾘﾝｸﾞ防止&Mark時間
            M += 1              #Mark長ｶｳﾝﾄｱｯﾌﾟ
        if CodeCount ==0:               #初めてMarkなら先頭ﾌﾗｸﾞ1
            CodeBuff=0b00000001
        if CodeCount < 7:               #7打点(6ｼﾌﾄの後+1ｶｳﾝﾄｱｯﾌﾟ)までなら(8打点以上は無視)
            CodeBuff <<= 1              #左ｼﾌﾄ(bit0に0が入る)
            CodeCount += 1              #Mark数ｶｳﾝﾄｱｯﾌﾟ
        if M >= D_len:                  #長(dash)短(dot)点判別
            CodeBuff |= 0b00000001#長点ならBit0=1(or 1),短点ならそのまま(bit0=0)
        S=0      #Space(打点間key=H)時間測定
        while (key_in.value !=1 and S<=D_len*4):#Spaceの間繰返し。長すぎる時抜ける
            time.sleep(0.01)    #10ms(変換時間と精度に影響。短→速,精度低下)
            S += 1                          #Space長ｶｳﾝﾄｱｯﾌﾟ
    return(CodeBuff)##Space=字間(3dot)以上の時,1字確定,関数抜ける

def CODEtoABC(b):       #CW_CODE → 欧文1文字に変換
    i=0
    for num in range(len(CWabc_CODE)):#欧文ｺｰﾄﾞﾃｰﾌﾞﾙの数,繰返し
        if CWabc_CODE[i] == b:          #ﾃｰﾌﾞﾙ参照
            return(abc_table[i])        #文字ﾃｰﾌﾞﾙから出力
        else:
            i += 1
    return('*')                         #該当なき場合,＊を返す

def CODEtoWA(b):        #CW_CODE → 和文1文字に変換
    i=0
    for num in range(len(CWwa_CODE)):#和文ｺｰﾄﾞﾃｰﾌﾞﾙの数,繰返し
        if CWwa_CODE[i] == b:   #ﾃｰﾌﾞﾙ参照
            return(wa_table[i]) #文字ﾃｰﾌﾞﾙから出力
        else:
            i += 1
    return('*')                         #該当なき場合,＊を返す

def ABCtoCODE(a):       #欧文半角大文字1字 → CW_CODEに変換
    i=0
    for num in range(len(abc_table)):#欧文ﾃｰﾌﾞﾙの数,繰返し
        if abc_table[i] == a:           #ﾃｰﾌﾞﾙ参照
            return(CWabc_CODE[i])       #文字ﾃｰﾌﾞﾙから出力
        else:
            i += 1
    return(0b01001100)          #該当なき場合,?「・・－－・・」を返す

    #CW送信 関連(BZ音,LED･･関連/dash,dot()で要出力設定)
CWtone = 700    #CW音周波数[Hz]設定
dot_len = 0.06  #CW送信速度(60ms=0.06s/短点dot[s])を決める
def dash():             #dash(長点)を作る関数/dash,dotで出力設定
    global CWtone
```

```
    global dot_len
    key_out.value=1     #H
    led1.on()
    beep(CWtone,dot_len*3)
    led1.off()
    key_out.value=0     #L

def dot():              #dot(短点)を作る関数
    global CWtone
    global dot_len
    key_out.value=1     #H
    led1.on()
    beep(CWtone,dot_len)
    led1.off()
    key_out.value=0     #L

def space_dd(): #1dot,wait
    global dot_len
    time.sleep(dot_len)

def CODEtoCW(a):        #CW_CODE→CW(CWｺｰﾄﾞを1字電信送信)
    Flag=0              #CW送出ﾌﾗｸﾞﾘｾｯﾄ
    if a == 0b10000000: #訂正符合<HH>「........」の時
        for num in range(8):    #dot*8
            dot()
            space_dd()
    else:
        for num in range(8):    #8回繰返し(先頭bit7～0まで順次)
            b=a & 0b10000000    #bit7=0ならb=0,bit7=1ならb=0x80
            if((b or Flag) != 0):#bit7=0,Flag=0なら抜ける
                if(b==0x80 and Flag==0):#bit7=1,Flag=0ならｶｳﾝﾄｱｯﾌﾟして抜ける
                    Flag += 1
                else:
                    if b != 0:  #bit7=1,Flag=1なら
                        dash()
                    else:               #bit7=0,Flag=1なら
                        dot()
                    space_dd()  #･-点間の空白1dotぶん待つ
            else:
                pass
            a <<= 1                     #左ｼﾌﾄ(次のbit)

def printCW(s): #欧文文字列→CW('･･'+chr(0x･･)を)送信
    l = len(s)  #引き渡された文字列の長さ(1～)をｶｳﾝﾄ
    i = 0
    for num in range(l):        #(1)～(l)回繰返し
        # 1文字送信
        if s[i] == ' ': #' 'なら単語間7dot(字間3と合わせ)
            space_dd()
            space_dd()
            space_dd()
            space_dd()
        else:
            b = ABCtoCODE(s[i]) #s[0]～s[l-1]
            CODEtoCW(b)
        #次の文字へ
        i += 1
        if i != (l-1):  #送信文字列の最後でないなら
            space_dd()  #文字間3dot(点間1dot含む)
            space_dd()
        else:                   #最後なので語間7dot(点間1dot含む)
            space_dd()
            space_dd()
            space_dd()
            space_dd()
            space_dd()
            space_dd()

def CWsong(s):#電信で歌 ('a4A')→melodyの音程･長さ+送信文字
    global SPEED        #ﾒﾛﾃﾞｨｰ速度0.05
    global CWtone       #700Hz
    global dot_len      #0.05s
    c=len(s)    #引き渡された文字列の長さ(1～)をｶｳﾝﾄ
    i=0
    for num in range(int(c*1/3)):       #文字数(1～)の3個1組回(整数)繰返し
        p=s[i]                                  #文字列i番目(音程)
        l=s[i+1]                                #次の数字(○分音符)
        a=s[i+2]                                #3番目(CW文字)
        if(ord(p)>=ord('a') and ord(p)<=ord('x')) :     #'a'～'x'のとき､ord()文字→ASCIIｺｰﾄﾞ
            CWtone=tone[ord(p)-ord('a')]        #音階ﾃｰﾌﾞﾙから周波数設定
```

```
            dot_len=(ord(l)-ord('0'))*SPEED/4    #音符長からCW速度設定
            b=ABCtoCODE(a)          #CW文字設定
            CODEtoCW(b)             #CW1字送信
            space_dd()              #間隔調整
            #space_dd()
        else:                                    #'a'～'x'以外のとき=休符
            time.sleep((ord(l)-ord('0'))*SPEED) #休符
        i += 3
    SPEED = 0.05          #ﾒﾛﾃﾞｨｰ速度設定,指定無き場合は0.05

#/**************************/
#/ main routine(ﾒｲﾝ関数)
#/**************************/
### initialize(初期設定)
### program(eternal loop → ^C stop!)
try:
    #「電信!ABCの歌」
    CWsong('d8A 1 d4B 2 k4C 2 k4D 2 m8E 8 m4F 2 k9G 9 ')
    CWsong('i5H 4 i8I 8 h3J 2 h4K 2 f4L 2 f5M 2 d8N 9 ')
    CWsong('k4O 2 k4P 2 i3Q 2 i4R 2 h8S 2 h8T 8 f4U 9 ')
    CWsong('k4V 2 k4W 2 i4X 2 i3Y 2 h9Z 2 f9? 9 ')
    melody('d4d4k4k4m4m4k4 4i4i4h4h4f4f4d8')
    #ﾒｯｾｰｼﾞ送信
    CWtone=700            #CW音程[Hz]設定(ｸﾞﾛｰﾊﾞﾙ変数)
    dot_len=0.06          #CW速度60ms(1dot長,ｸﾞﾛｰﾊﾞﾙ変数)
    printCW('  SONG BY JA1YZT ')#<AR>は<,A,R,>と読んでしまう‥
    y=ABCtoCODE('<AR>') #<AR>(本文終了)を1ｺｰﾄﾞにﾃｰﾌﾞﾙ変換
    CODEtoCW(y)           #1ｺｰﾄﾞを送信

    while True:
        #CW解析
        b = CWtoCODE()  #CWin→2進ｺｰﾄﾞへ
        a = CODEtoABC(b)#ｺｰﾄﾞ→欧文1文字
        w = CODEtoWA(b) #ｺｰﾄﾞ→和文1文字
        CODEtoCW(b)               #ｺｰﾄﾞ→CWout
        print(f'D_len={D_len},CodeBuff={bin(b)},欧文:{a},和文:{w}')

    #CW送信練習ﾌﾞｻﾞｰ
#    while True:
#        if key.value==1:
#            led1.on()
#            beep(800,0.02)
#        else:
#            #beep(0,0)
#            led1.off()

except KeyboardInterrupt:       #^Cで終了処理へ
    pass
### to end(ﾌﾟﾛｸﾞﾗﾑ終了時実行処理)
led1.close()
bz.close()
key_in.close()
key_out.close()

#/資料：-------------------------
#/「RaspberryPiで学ぶ電子工作」金丸隆志
#/                    (技術評論社2024)
#/「ｴﾚｷｼﾞｬｯｸ20号」PIC記事 光永法明
#/      (CQ出版)       →ﾌﾞｻﾞｰ関連(JA1YEF/2013)
#/「PC-6001によるﾓｰﾙｽ信号の解読実験ﾌﾟﾛｸﾞﾗﾑ」JI1NZL/青木昇
#/ (CQﾊﾑﾗｼﾞｵ1991.4/CQ出版)Basicｱﾙｺﾞﾘｽﾞﾑ+ﾈｯﾄ詠み人知らず
#/ →C言語/PIC12F1822(+改)  (JA1YEF/日立一高物理部2013.3)
#/ →python/RasPi5(+和,song)(JA1YZT/太田一高物理部2026.3)
#/-----------------------------


```