

Savoir-faire : Définir la structure matérielle d'un produit en fonction des caractéristiques techniques.

Problématique : Des informations sont transmises par liaison UART. Vous devez décoder ces informations en écoutant la ligne.

Les documents ressources sur la liaison série synchrone UART et sur l'ESP32 sont à consulter pour réaliser l'activité.

1. Le protocole UART

Q1 : Que veut dire UART ?

Q2 : Expliquer pourquoi la liaison UART est une liaison asynchrone.

Q3 : Indiquer de combien de liaison série UART dispose l'ESP32 ?

Q4 : Donner les broches du GPIO correspondantes.

Le signal suivant a été capté sur la liaison UART. Huit bits ont été transmis. La durée d'un bit est de 104µs.

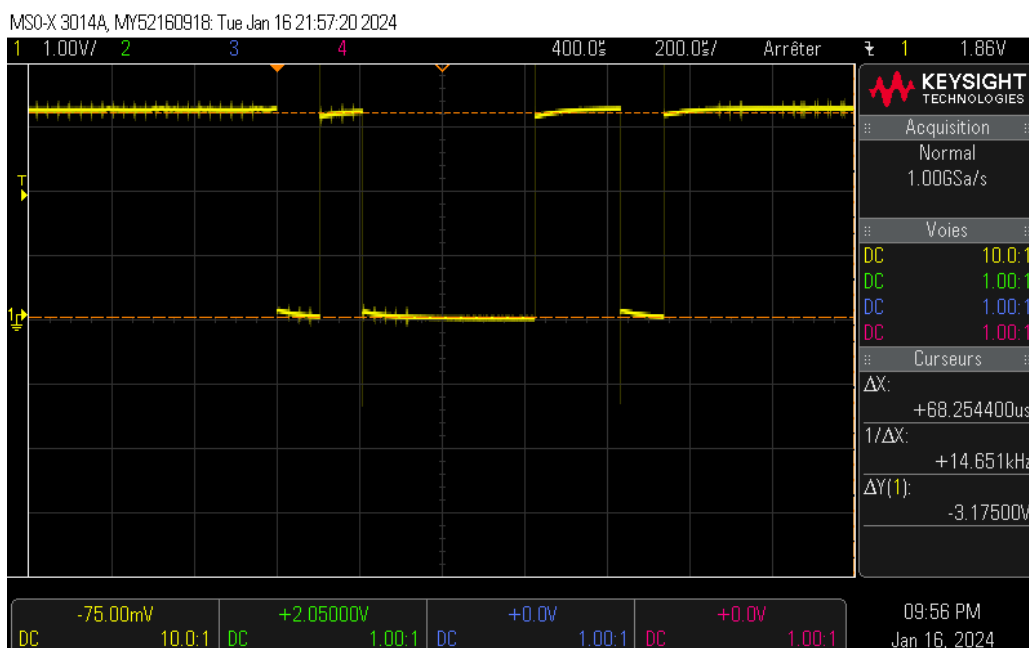


Figure 1 : signal capté sur une liaison UART

Q5 : Décoder la trame ci-dessus (figure 1) et donner la valeur binaire puis la valeur hexadécimale transmise.

Q6 : La donnée transmise est un caractère ASCII, retrouver ce caractère à l'aide du document technique DT1.

Q7 : Calculer la vitesse de transmission en bits par seconde (baudrate).

Un autre signal, composé de plusieurs caractères a été capté sur la liaison UART. Le même protocole a été utilisé que précédemment.

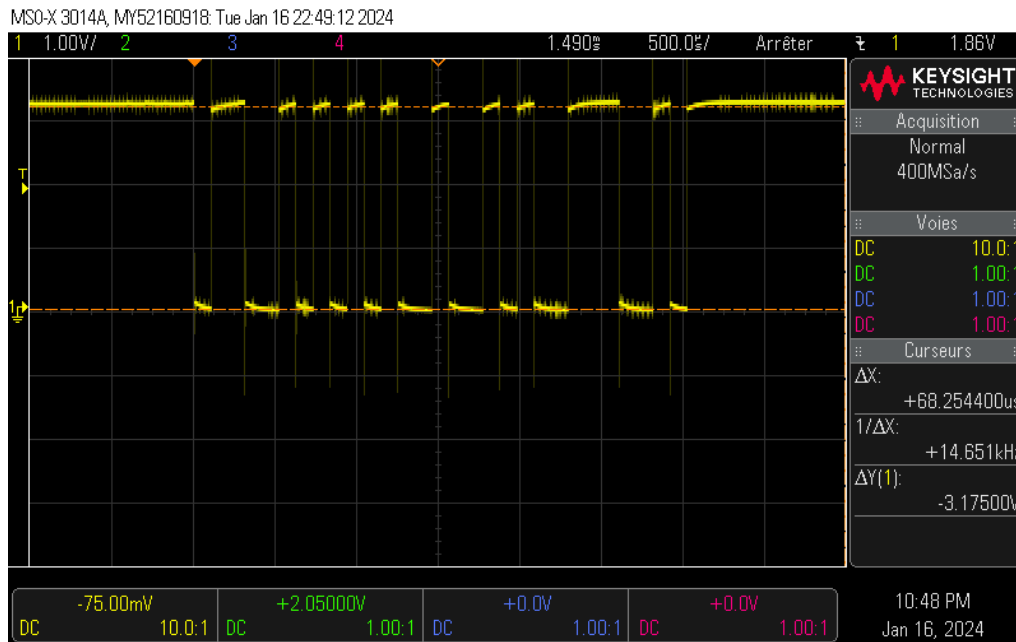


Figure 2 : signal capté sur une liaison UART

Q8 : Décoder la trame ci-dessus (figure 2) et donner retrouver la chaîne de caractères transmise par la liaison UART.

2. Captage des signaux de la liaison UART

Q9 : Afin de capturer le signal émis par une des liaisons UART du module ESP32, indiquer quelle broche faut-il observer à l'oscilloscope (RX ou TX) ? Justifier la réponse.

☞ Connecter l'oscilloscope à la broche choisie précédemment et relier la masse de l'oscilloscope à la borne GND du module ESP32.

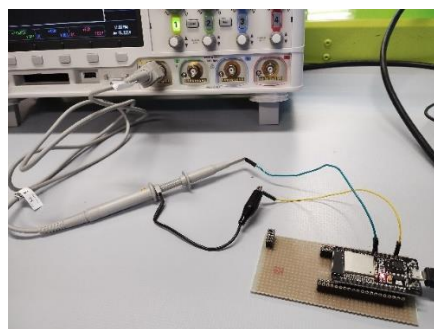


Figure 3 : montage pour visualiser la trame UART issue de l'ESP32

☞ Exécuter le programme ci-dessous et capter le signal à l'oscilloscope.

```
from machine import UART

# configuration de la liaison UART en fonction des broches RX et TX
uart = UART(1, baudrate=9600, tx=17, rx=16)
uart.init(9600, bits=8, parity=None, stop=1)

uart.write('a')
```

☞ Observer le signal à l'oscilloscope et vérifier que le signal reçu correspond à la lettre a.

3. Bit de parité

Les signaux ci-dessous correspondent tous les deux à la transmission du caractère « a ». L'un a été transmis en utilisant le protocole incluant la parité paire et l'autre la parité impaire.

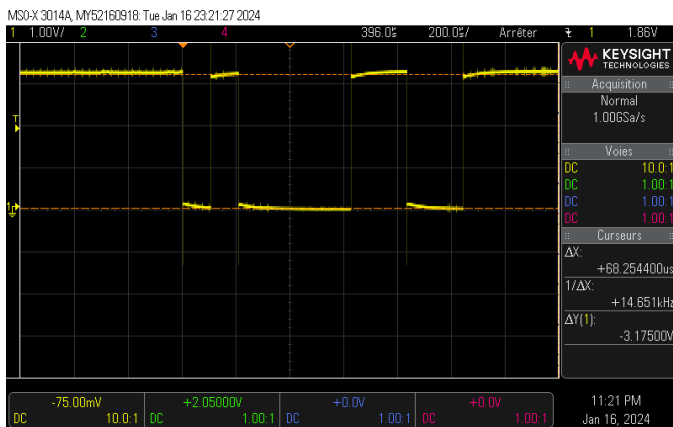


Figure 4

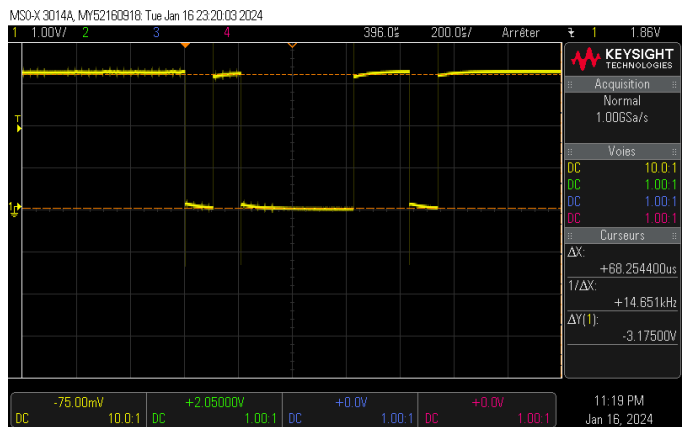


Figure 5

Q10 : Indiquer pour chaque signal ci-dessus (figure 4 et 5) si le protocole utilise la parité paire ou impaire. Justifier la réponse.

Q11 : Modifier le programme pour envoyer la lettre B en parité paire.

☞ Observer le signal à l'oscilloscope et vérifier que le signal reçu correspond à la lettre B.

4. Décodage de la trame par l'oscilloscope

En indiquant le protocole utilisé, l'oscilloscope est capable de décoder la trame reçue.

✎ Envoyer sur la liaison UART le mot STI2D en parité impaire et utiliser le bouton « Serial » pour décoder la trame à l'aide de l'oscilloscope.

- Pour ceci configurer correctement le nombre de bits, la parité et la vitesse de transmission et l'ordre des bits à partir du menu **Config bus**. Placer la polarité en **Inact**.

✎ Vérifier la transmission du mot STI2D.

DT1 : table ASCII

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21		65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Figure 6 : table ASCII