EEE3453 Computer Programming for Microcontroller
End-of-Module Assessment

Instructions to Students:

(1) The EEE3453 End-of-Module Assessment (EA) is a **project assessment**. **Plagiarism will be regarded as cheating.** All *misbehavioural cases* will be reported to the Head of Department for disciplinary action.

(2) The EA is run as a 2-stage Mini Project.

(3) Students have to
- carefully study this project statement
- design and implement the programs as required
- follow the due date of submission to submit the program specified in each stage to the EEE3453 Moodle website accordingly
- Penalties of marks shall be applied to late submission

(4) Schedule of Submissions:

Activities	Mark Distribution	Date
Submission of Stage-1 Work (**Softcopy of program, BOTH hardcopy and softcopy of MS Word file**)	50%	1A: 9:00 on 18/12/2023 (Mon) 1B: 15:00 on 19/12/2023 (Tue) 1C: 9:00 on 19/12/2023 (Tue)
Submission of Stage-2 Work (**Softcopy of program, BOTH hardcopy and softcopy of MS Word file**)	50%	1A: 9:00 on 11/12/2023 (Mon) 1B: 15:00 on 12/12/2023 (Tue) 1C: 9:00 on 12/12/2023 (Tue)

Stage-1 work (50%) – Individual
Stage-2 work (50%) – Group (3 to 4 students per group); Report (25%);
Demonstration – viva and demonstration to lecturer (25%)

(5) Students have to return the ESP32 and associated accessories to lab on or before Week 15.

Note: The submission schedules are for reference; lecturers may adjust the timeframe as long as the task assessments are completed.

EEE3453 Mini Project

Purpose

An Excel file (filename: ***TestData.xlsx***) is available for download from the EEE3453 Moodle website.

Assessment

- The mini project involves 2 stages with 2 programs, 2 reports, and a practical demonstration/viva. Students need to develop the **programs for Stage 1 individually** and upload the programs of each stage to the Moodle platform on time.
- The program design and coding should be systematic and structural.

  Declaration of variables and constants in a meaningful manner
  Block organization and readability of source program (e.g. indentation)
  Annotations or comment added to the program, etc.
- **No re-submission** to Moodle is allowed. Test the programs thoroughly before each submission.

Stage 1

Filename: 1X_YourName_Stage1.ino (e.g. 1A_ChanTaiMan_Stage1.ino)

Filename: 1X_YourName_Stage1-result.docx (e.g. 1A_ChanTaiMan_Stage1-result.docx)

Each student submits **1 program file (softcopy) and 1 MS Word file (BOTH hardcopy and softcopy)**.

Due Date of Submission: Refer to the schedule on p.1

Write a program to **Read String (a sequence of characters) from Serial Monitor Input.**

Capture all the result output to an MS Word file (1X_YourName_Stage1-result.docx).

*Remember to select "**Newline**" at the bottom of the Serial Monitor in the Arduino IDE*

Note: To execute the 'Copy-and-paste' operation under the Serial Monitor, carry out the following steps in sequence:

(1) Select and copy a block of data from the Excel file by pressing Ctrl-C.

(2) Click the 'Send Data' textbox and then press Ctrl-V. This step will paste the selected Excel data into the 'Send Data' textbox.

(3) Hit the [Send] button to send the data to the ESP32.

Functionality:

- Define a **char array** of 1200 bytes to store the input data.
- Store the input to the array.
- Limit the input to be within the array size, i.e. store only the first 1200 characters of data and ignore all others.
- Firstly, display the size of the input data in the Serial Monitor.
- Then, display the input data nicely in the Serial Monitor.
- The program should be able to accept new input again.

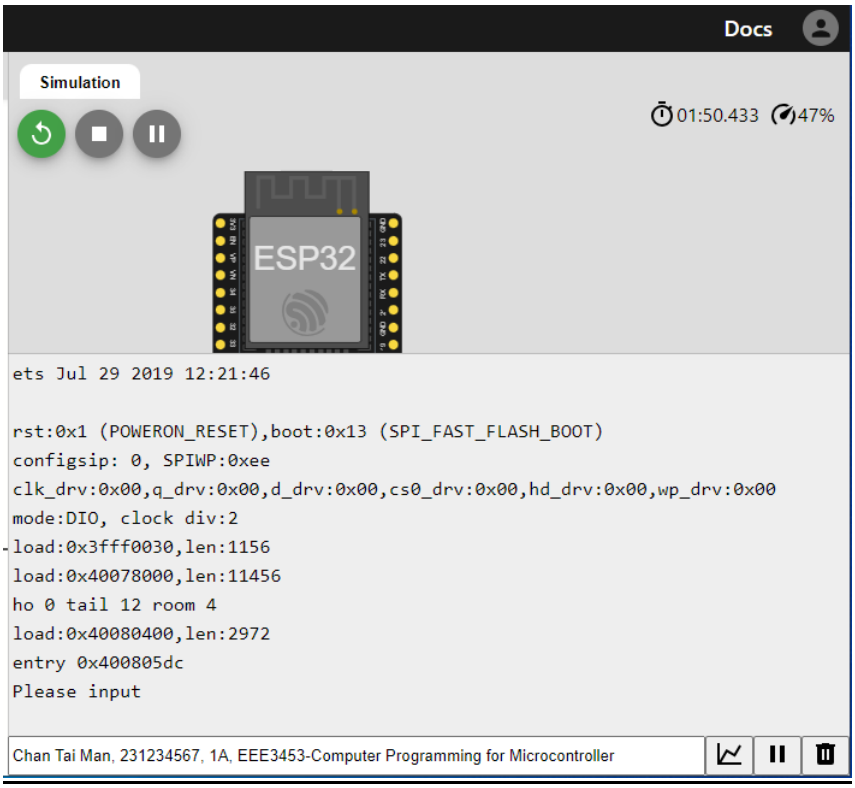
Functionality test:

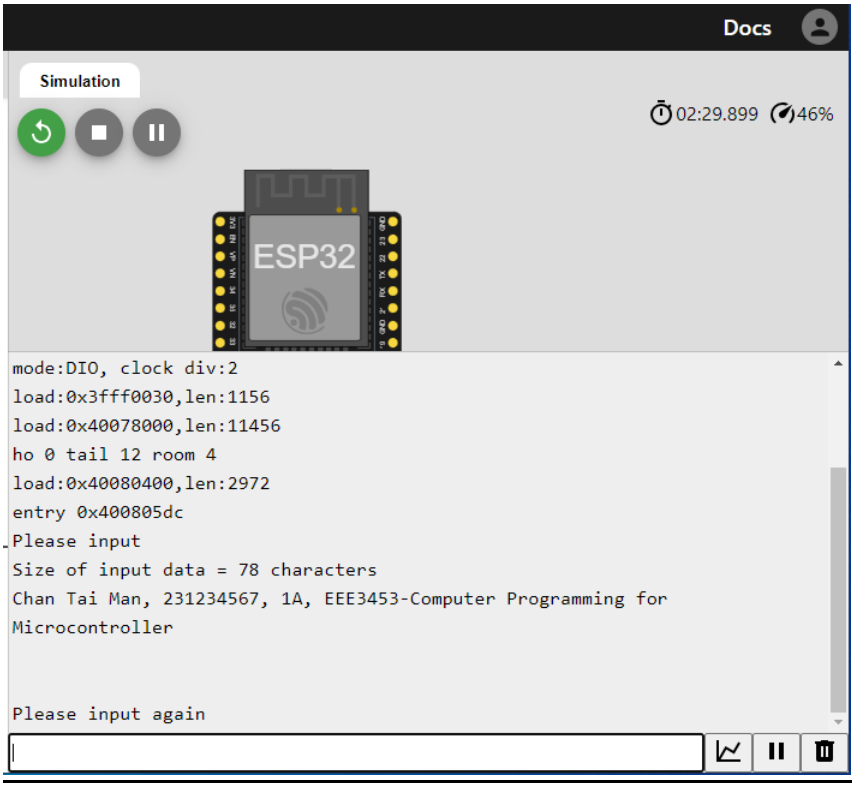
- Input your student details including English full name, student id, class name, and 'EEE3453-Computer Programming for Microcontroller' in the Serial Monitor. (e.g. Chan Tai Man, 231234567, 1A, EEE3453-Computer Programming for Microcontroller).
- Display the size of input data and then the data itself in the Serial Monitor.
- Copy and paste a random set of data (with a size less than 1200 characters) from 'Column A' in the 'Random string data' worksheet of the Excel file to the Serial Monitor.
- Display the size of input data and then the data itself in the Serial Monitor.
- Copy and paste another random set of data (with a size more than 1200 characters) from 'Column A' in the 'Random string data' worksheet of the Excel file to the Serial Monitor.
- Display the size of input data and then the data itself in the Serial Monitor.
- The program should be able to accept new input.

Upload the finished program to Moodle.

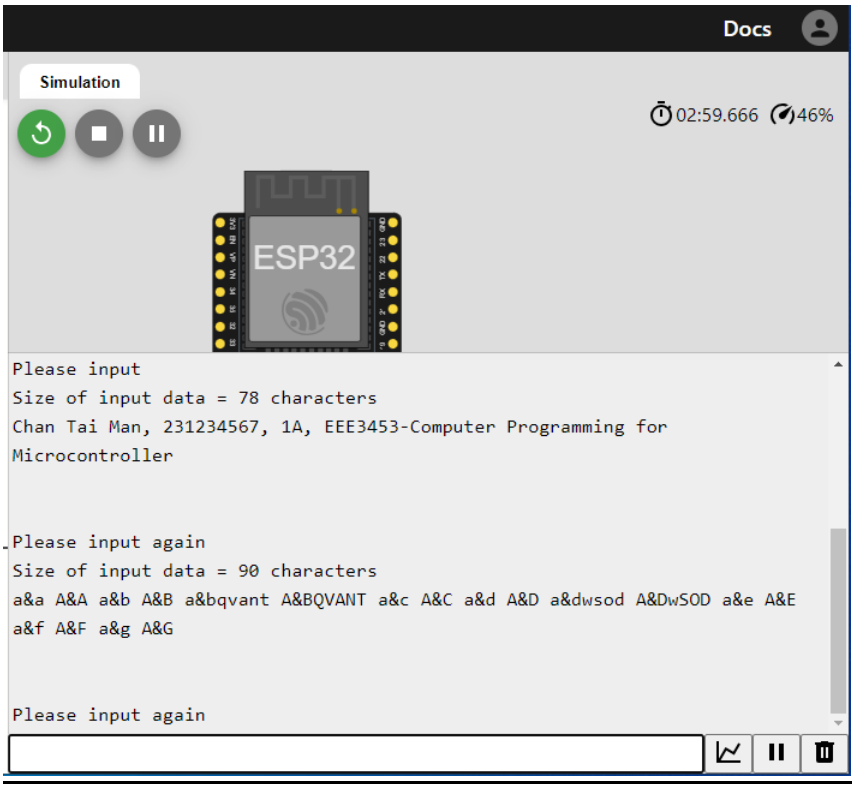
Sample output of Stage-1 program:

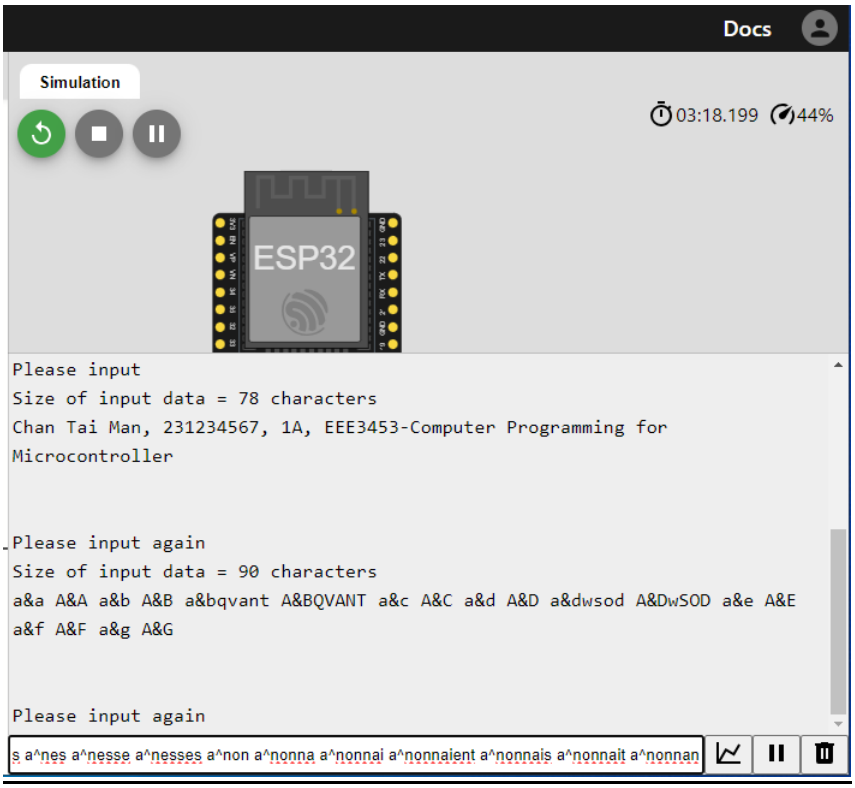
Wokwi version

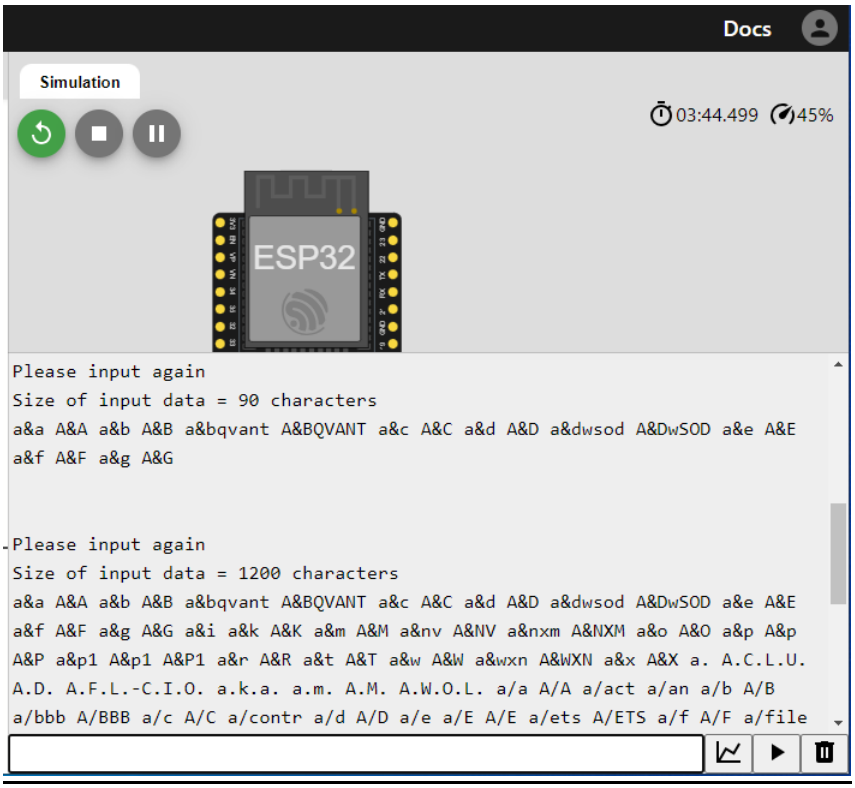


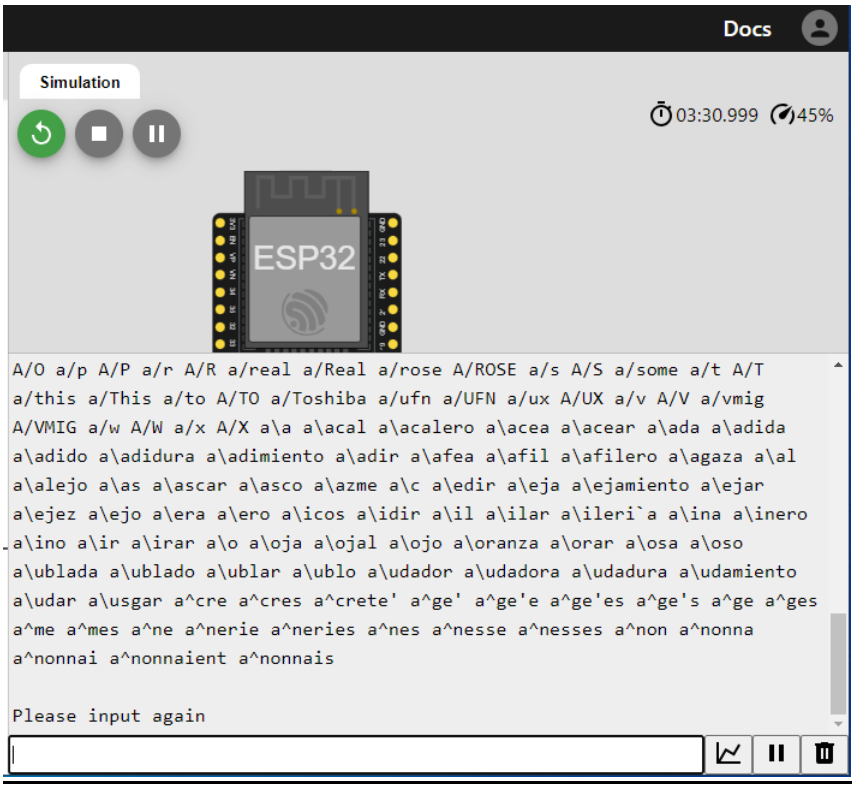



Docs
Simulation
02:49.133
46%
ESP32
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
Please input
Size of input data = 78 characters
Chan Tai Man, 231234567, 1A, EEE3453-Computer Programming for
Microcontroller
Please input again
A&B a&bqvant A&BQVANT a&c A&C a&d A&D a&dwsod A&DwSOD a&e A&E a&f A&F a&g A&G

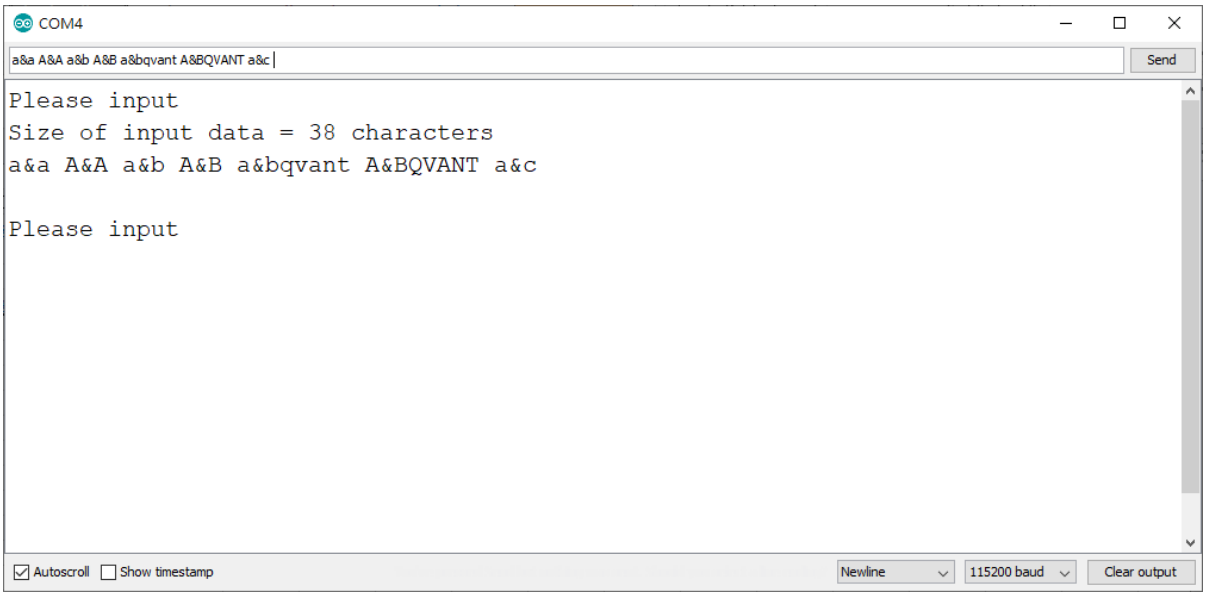

Docs
Simulation
02:59.666
46%
ESP32
Please input
Size of input data = 78 characters
Chan Tai Man, 231234567, 1A, EEE3453-Computer Programming for
Microcontroller
Please input again
Size of input data = 90 characters
a&a A&A a&b A&B a&bqvant A&BQVANT a&c A&C a&d A&D a&dwsod A&DwSOD a&e A&E
a&f A&F a&g A&G
Please input again

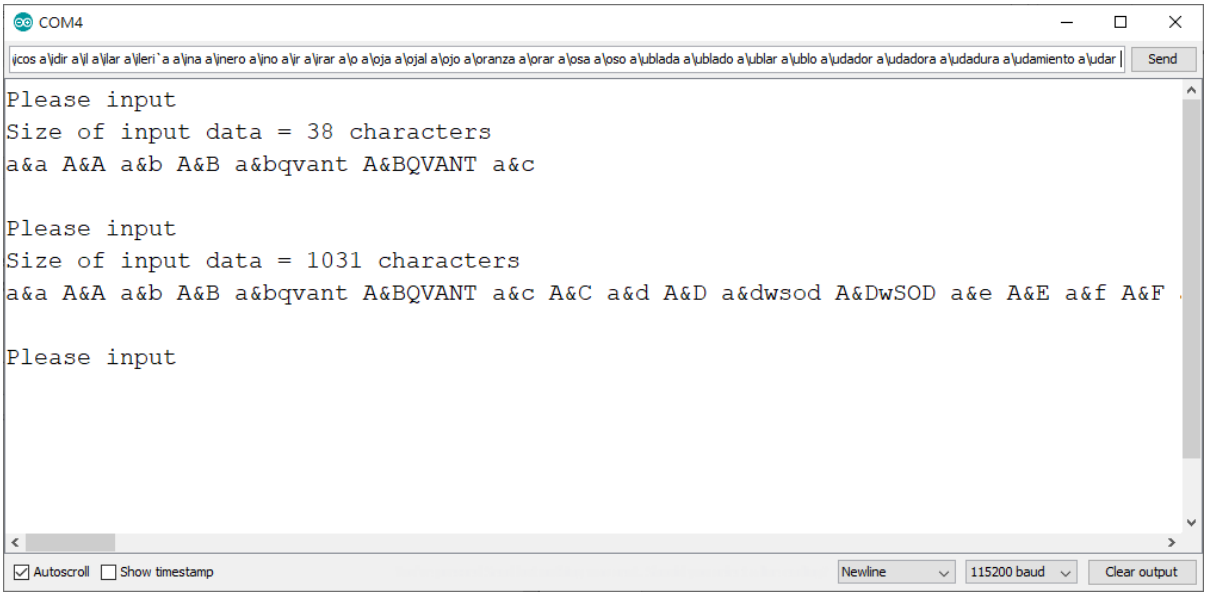

Docs
Simulation
03:18.199
44%
ESP32
Please input
Size of input data = 78 characters
Chan Tai Man, 231234567, 1A, EEE3453-Computer Programming for
Microcontroller
Please input again
Size of input data = 90 characters
a&a A&A a&b A&B a&bqvant A&BQVANT a&c A&C a&d A&D a&dwsod A&DwSOD a&e A&E
a&f A&F a&g A&G
Please input again
s a^nes a^nesse a^nesses a^non a^nonna a^nonnai a^nonnaient a^nonnais a^nonnait a^nonnan

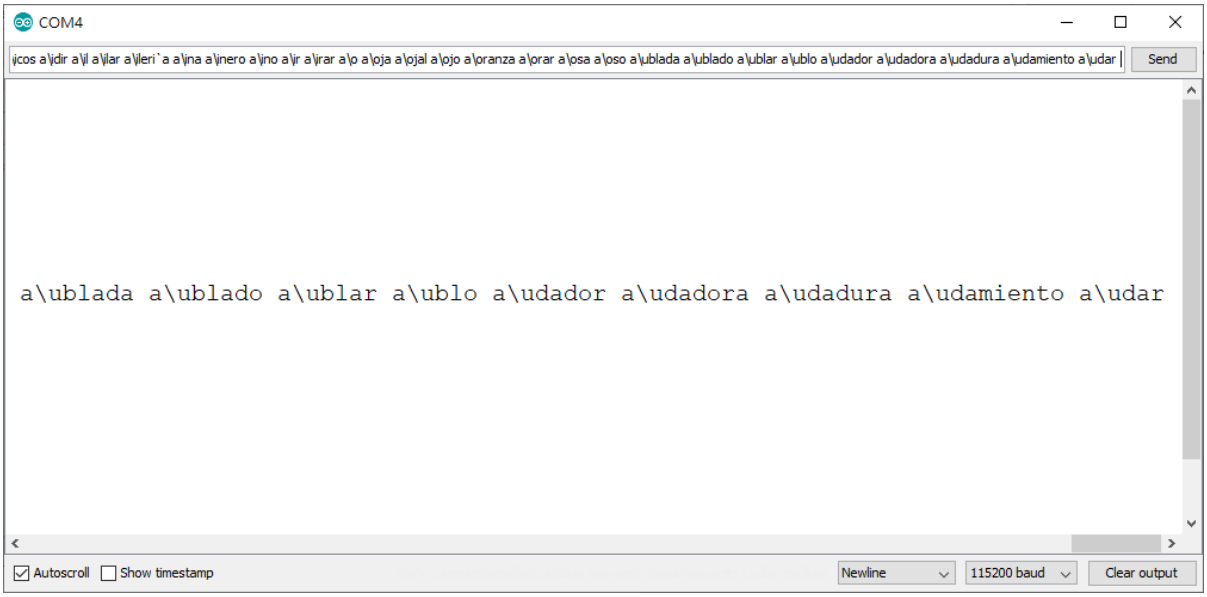

Docs
Simulation
03:44.499
45%
ESP32
Please input again
Size of input data = 90 characters
a&a A&A a&b A&B a&bqvant A&BQVANT a&c A&C a&d A&D a&dwsod A&DwSOD a&e A&E
a&f A&F a&g A&G
Please input again
Size of input data = 1200 characters
a&a A&A a&b A&B a&bqvant A&BQVANT a&c A&C a&d A&D a&dwsod A&DwSOD a&e A&E
a&f A&F a&g A&G a&i a&k A&K a&m A&M a&nv A&NV a&nxm A&NXM a&o A&O a&p A&p
A&P a&p1 A&p1 A&P1 a&r A&R a&t A&T a&w A&W a&wxn A&WXN a&x A&X a. A.C.L.U.
A.D. A.F.L.-C.I.O. a.k.a. a.m. A.M. A.W.O.L. a/a A/A a/act a/an a/b A/B
a/bbb A/BBB a/c A/C a/contr a/d A/D a/e a/E A/E a/ets A/ETS a/f A/F a/file

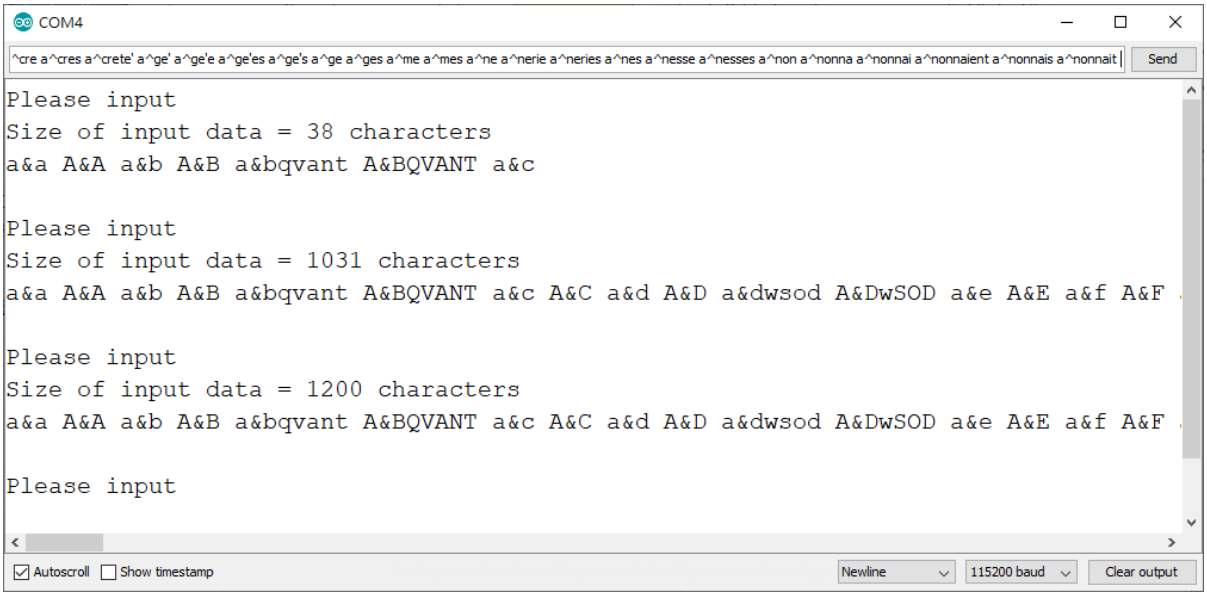

Docs
Simulation
03:30.999
45%
ESP32
A/O a/p A/P a/r A/R a/real a/Real a/rose A/ROSE a/s A/S a/some a/t A/T
a/this a/This a/to A/TO a/Toshiba a/ufn a/UFN a/ux A/UX a/v A/V a/vmig
A/VMIG a/w A/W a/x A/X a\a a\acal a\acalero a\acea a\acear a\ada a\adida
a\adido a\adidura a\adimiento a\adir a\afea a\afil a\afilero a\agaza a\al
a\alejo a\as a\ascar a\asco a\azme a\c a\edir a\eja a\ejamiento a\ejar
a\ejez a\ejo a\era a\ero a\icos a\idir a\il a\ilar a\ileri`a a\ina a\inero
a\ino a\ir a\irar a\o a\oja a\ojal a\ojo a\oranza a\orar a\osa a\oso
a\ublada a\ublado a\ublar a\ublo a\udador a\udadora a\udadura a\udamiento
a\udar a\usgar a^cre a^cres a^crete' a^ge' a^ge'e a^ge'es a^ge's a^ge a^ges
a^me a^mes a^ne a^nerie a^neries a^nes a^nesse a^nesses a^non a^nonna
a^nonnai a^nonnaient a^nonnais
Please input again

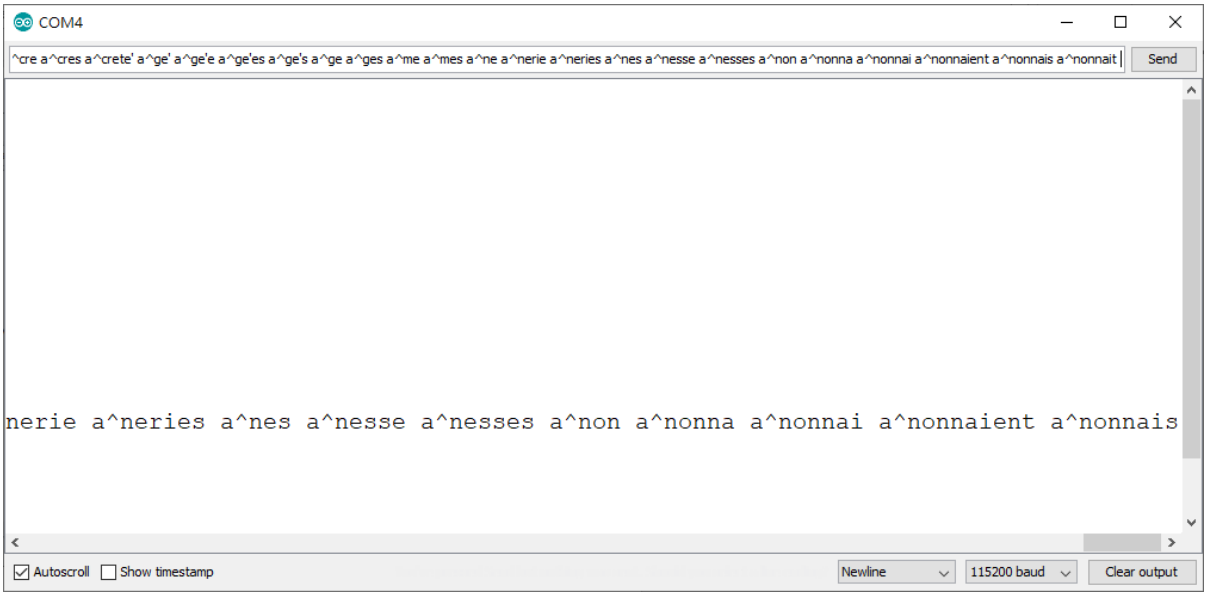
Arduino IDE version








COM4
^cre a^cres a^crete' a^ge' a^ge'e a^ge'es a^ge's a^ge a^ges a^me a^mes a^ne a^nerie a^neries a^nes a^nesse a^nesses a^non a^nonna a^nonnai a^nonnaient a^nonnais a^nonnait
Send
Please input
Size of input data = 38 characters
a&a A&A a&b A&B a&bqvant A&BQVANT a&c
Please input
Size of input data = 1031 characters
a&a A&A a&b A&B a&bqvant A&BQVANT a&c A&C a&d A&D a&dwsod A&DwSOD a&e A&E a&f A&F
Please input
Size of input data = 1200 characters
a&a A&A a&b A&B a&bqvant A&BQVANT a&c A&C a&d A&D a&dwsod A&DwSOD a&e A&E a&f A&F
Please input
Autoscroll
Show timestamp
Newline
115200 baud
Clear output


COM4
^cre a^cres a^crete' a^ge' a^ge'e a^ge'es a^ge's a^ge a^ges a^me a^mes a^ne a^nerie a^neries a^nes a^nesse a^nesses a^non a^nonna a^nonnai a^nonnaient a^nonnais a^nonnait
Send
nerie a^neries a^nes a^nesse a^nesses a^non a^nonna a^nonnai a^nonnaient a^nonnais
Autoscroll
Show timestamp
Newline
115200 baud
Clear output

Stage 2

Filename: 1X_GpY_Stage2.ino (e.g. 1A_Gp1_Stage2.ino)

Filename: 1X_GpY_Stage2-report.docx (e.g.: 1A_Gp1_Stage2-report.docx)

Each group submits only **1 program file (softcopy) and 1 MS Word file (BOTH hardcopy and softcopy)**.

Due Date of Submission: Refer to the schedule on p.1

You are required to create an application utilizing ESP32 microcontroller, sensor technologies, and smart technologies. The application should have practical real-life applications and can be developed using either simulation software (Wokwi) or a physical circuit (Arduino IDE). Some examples of potential applications include but are not limited to, temperature and humidity detection for triggering fire-alarm systems, controlling flashing lights for road traffic, motor control for flying drones, fans, or vacuum cleaners, and light detection for anti-theft systems. Please write a report for your application, including the schematic circuit and output results, and save it as an MS Word file named "1X_GpY_Stage2-report.docx".

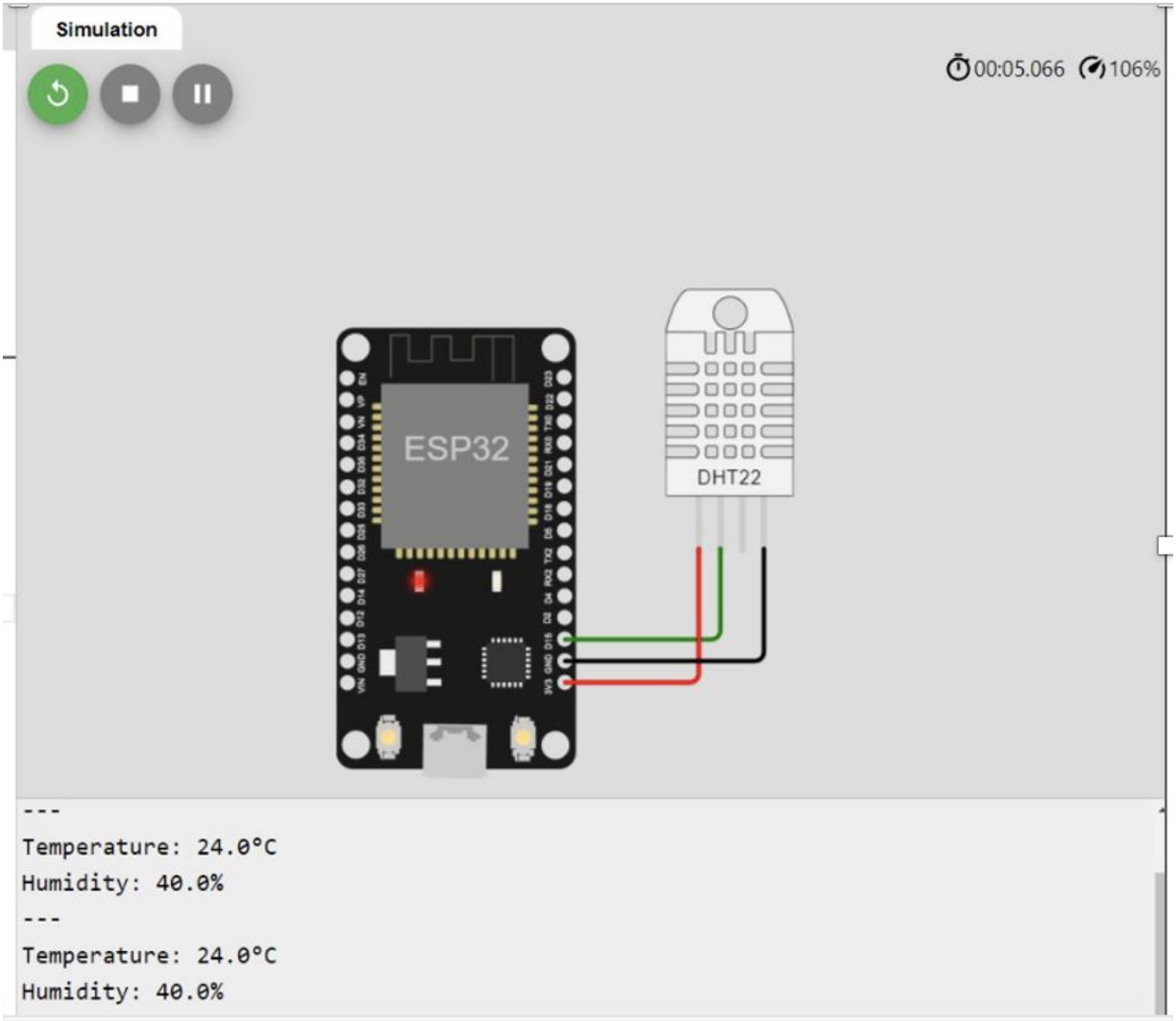
Marks component

Application	**Component**
Report (25%)	Description (10%)
	Schematics (5%)
	Results (5%)
	Conclusion (5%) (**Individual**)
Demonstration (25%)	Functionality (8%)
	Coding (7%)
	Viva (10%) (**Individual**)

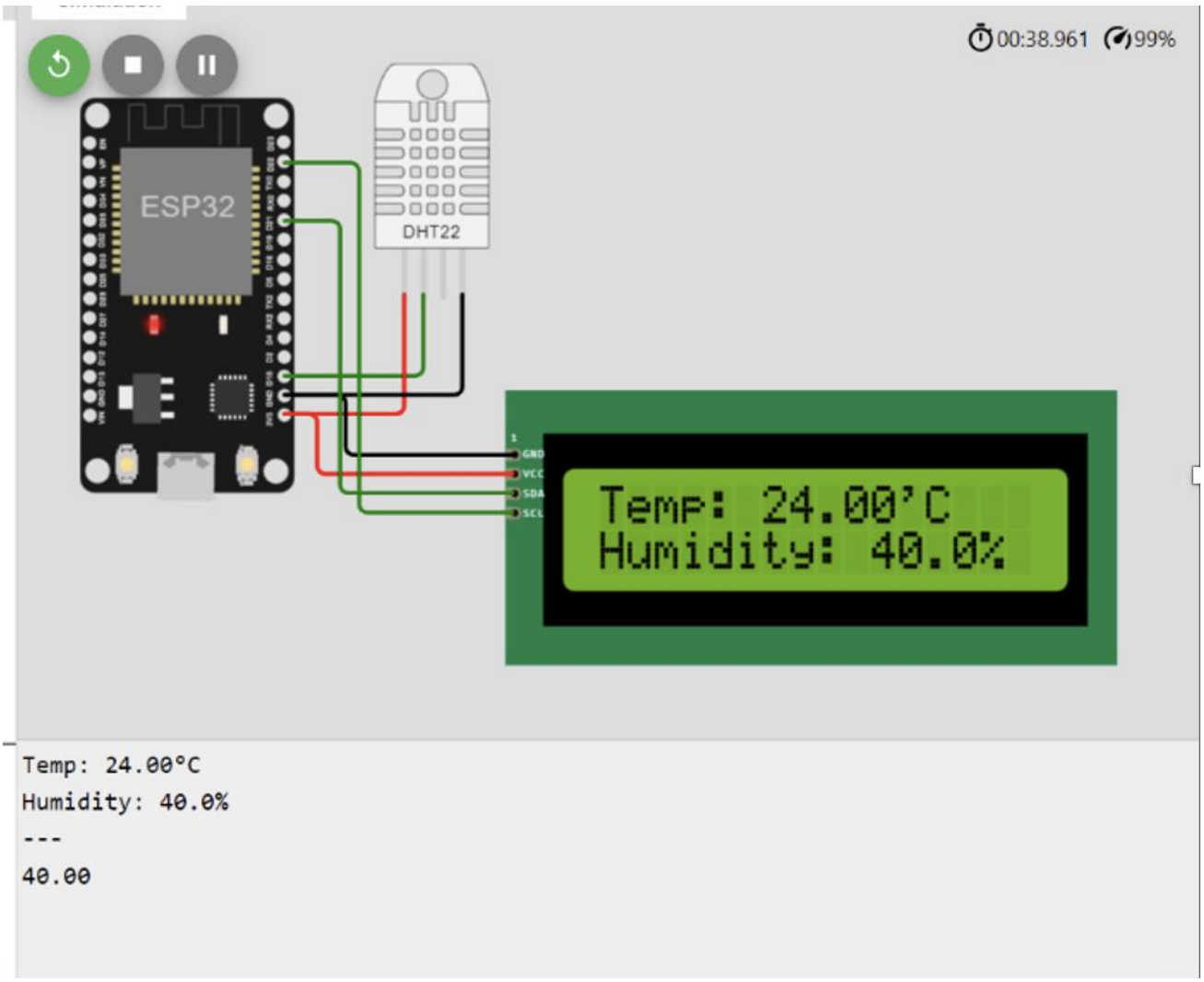
Upload the finished program to Moodle.

Sample output of Stage-2 program:

Sample 1



Sample 2




```
Temp: 24.00°C
Humidity: 40.0%
---
40.00
```

ASCII Table

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	Del

asciicharstable.com

**** * * End * * ****