MULTI-LINGUAL SPEECH TO BRAILLE CONVERTER USING ARDUINO UNO AND IoT

Under the Supervision

of

B.N.S. Munda

(Associate Professor

Electronics and Communication Engineering)

by

Akshita Shrivastava (2021UGEC114)

Ayush Kumar Lal (2021UGEC090)

Khushal Kumar Rajak (2021UGEC080)

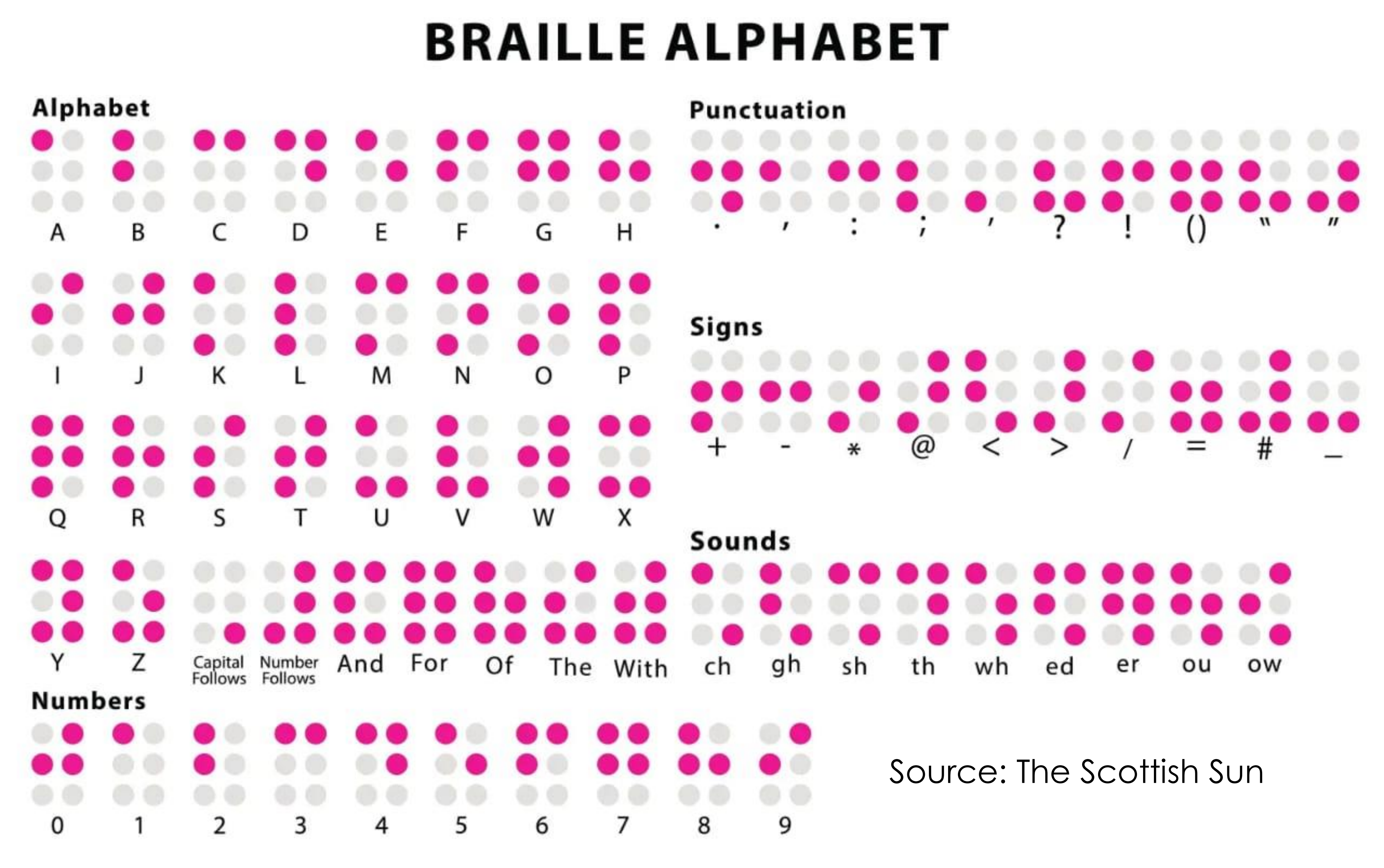
Gipashu Aryan (2021UGEC079)

Likhit Reddy (2021UGEC078)

Pragati Dubey (2021UGEC008)

INTRODUCTION

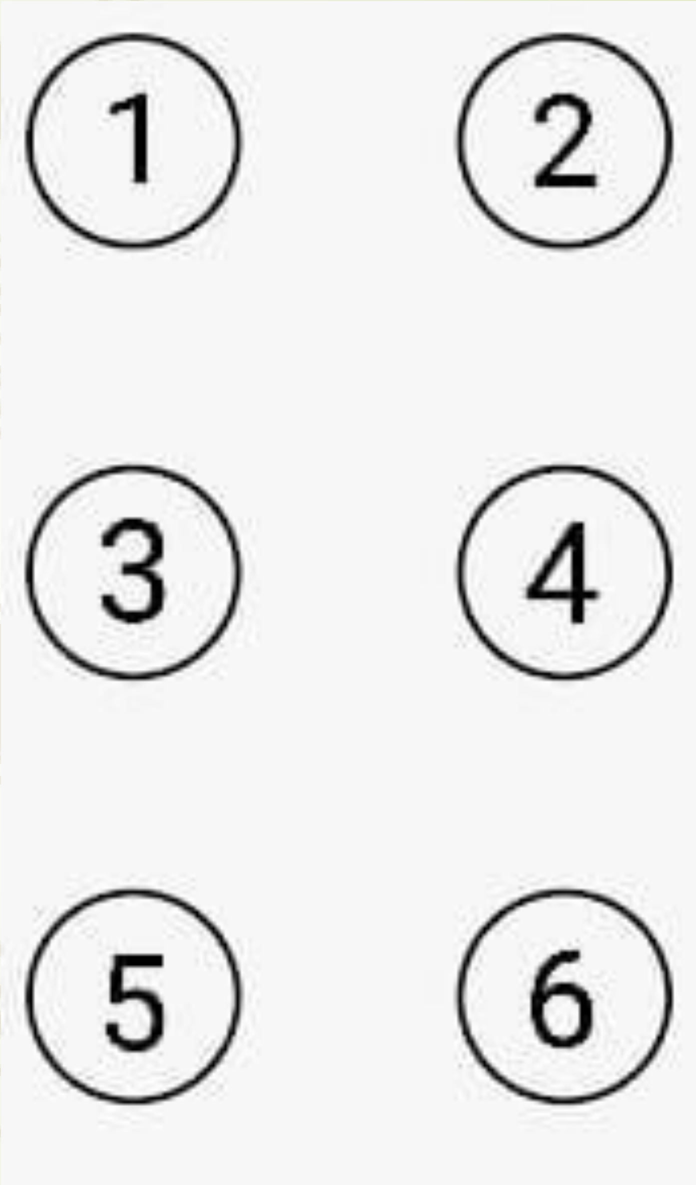
- **The basic Braille system contains 6 raised dots to create a possible combination of 64 different signs.**
- **The visually challenged can use the raised dots of to detect the Braille codes.**
- **The primary focal point of the project is to upgrade the pace of education and to make visually impaired people self-sufficient in education**



Source: The Scottish Sun

THE BRAILLE CELL

- It consists of 6 different cells arranged row-wise
- LEDs/x-axis motor can be used in place of each cell to read a letter
- The arrangement of the LEDs are kept in order with the cell numbers as shown in the figure



Self-made png

MOTIVATION

- 43 million people are living with severe blindness and 295 million people living with moderate-to-severe blindness in the world.

- Nearly 0.2% of the world’s population – 16 million people are living with severe deaf-blindness and around 2% of people – 160 million suffer from acute deafness or blindness of some sort.

- In India, there are about 1 million cases of deaf-blindness alone.

MOTIVATION

- Accessibility
- Cost-effectiveness
- Multilingual Support
- Educational Tool
- Reusability
- Scalability

OBJECTIVES

1. The Phase-1 objective is to develop a model for converting text in English to Braille script.

2. Phase-2 includes upgrading the previous work and adding speech support to the initial model.

3. The first and second phases of work will be integrated to form a fully functional speech-to-Braille converter.

4. The same model can then be expanded to multiple languages, making the model more effective and accessible. Languages like Hindi can be used as input.

LITERATURE STUDY

- Arduino UNO Documentation: Language, function, variables, structures, libraries, etc.

- National Braille Press Website: Braille books and literacy mission.

- Research Papers:
- History and Development in Technology

The American Printing House for the Blind in Louisville installed an IBM 709 computer that produced Grade 2 Braille on punched cards, which were then used to drive stereotyping machines.

LITERATURE STUDY

- One system uses electromagnetic solenoids and an Arduino Uno to display Braille, while another system employs rotary actuators and stepper motors controlled by an 8051 microcontroller to manipulate eight pins in each Braille cell.

- All the systems were very complicated, non-cost-efficient, and not so user friendly.

WORK DONE SO FAR

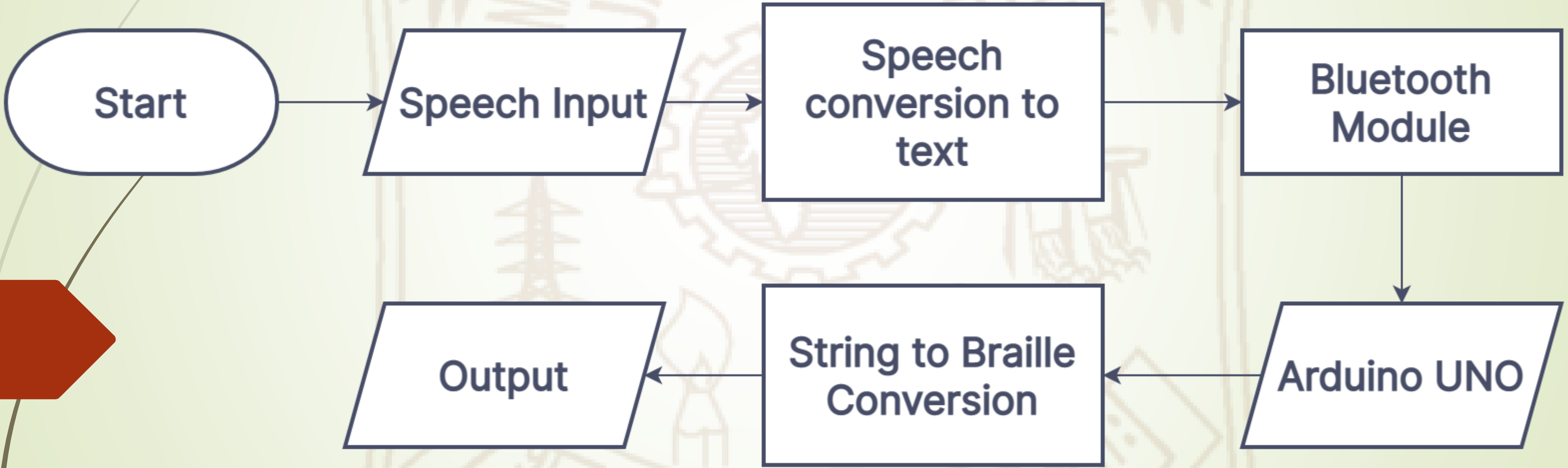
1ST PHASE:

The first phase includes a basic braille translator. Its primary aim is to convert sentences/strings into Braille script using motors.

2ND PHASE:

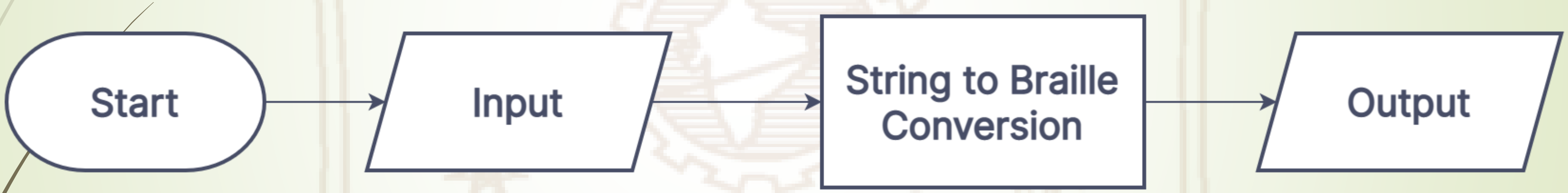
It will include speech recognition and text conversion. This will be achieved using a Bluetooth module which will be integrated with the basic braille translator.

METHODOLOGY



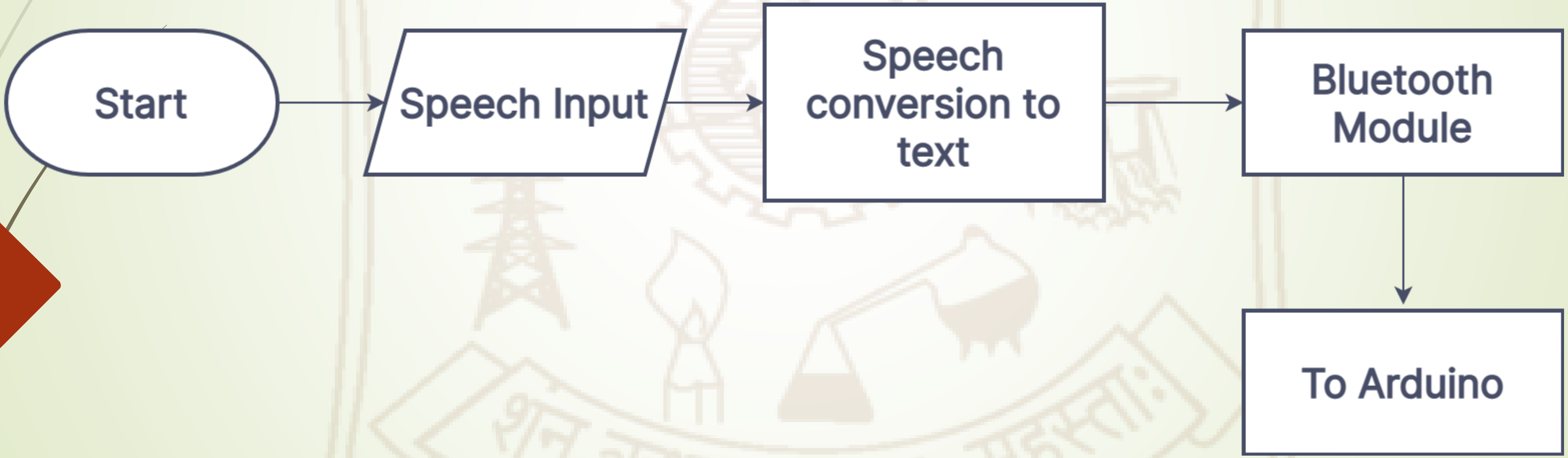
METHODOLOGY

1ST PHASE:



METHODOLOGY

2ND PHASE:



RESULTS & DISCUSSION

TEXT TO BRAILLE CODE IMPLEMENTATION:

- Declare character array with one index for every possible ASCII byte/character.
- Set pin 2 of Arduino for the least significant bit and pin 3, 4, 5, 6, and 7 subsequently for higher bits and pin 8 for the buzzer for a non-translatable character.
- If any non-translatable character comes then the buzzer LED blinks 2 times.

RESULTS & DISCUSSION

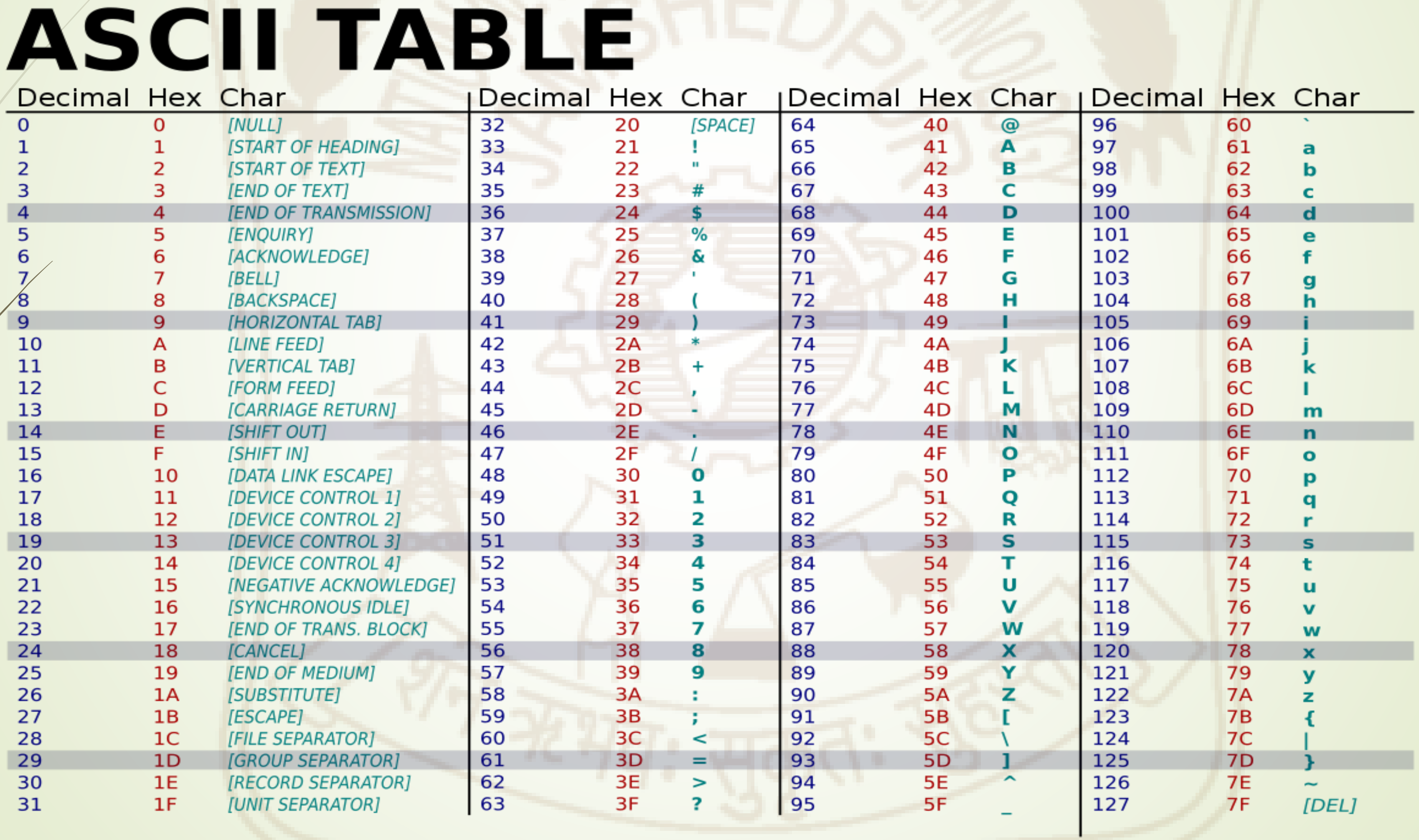
- Code reads 9600 bits per second as it is the standard number to work with Arduino.

- Input is through the serial monitor. Serial.available() checks whether the input is given if the input is given then the loop starts.

- Input is converted to the ASCII number of that character, default initialization is 99 for a non-translatable character and the buzzer LED blinks when it is detected.

- Starting pin is 2 in this loop and bit masking starts from 000001and shifts left on each iteration.

RESULTS & DISCUSSION

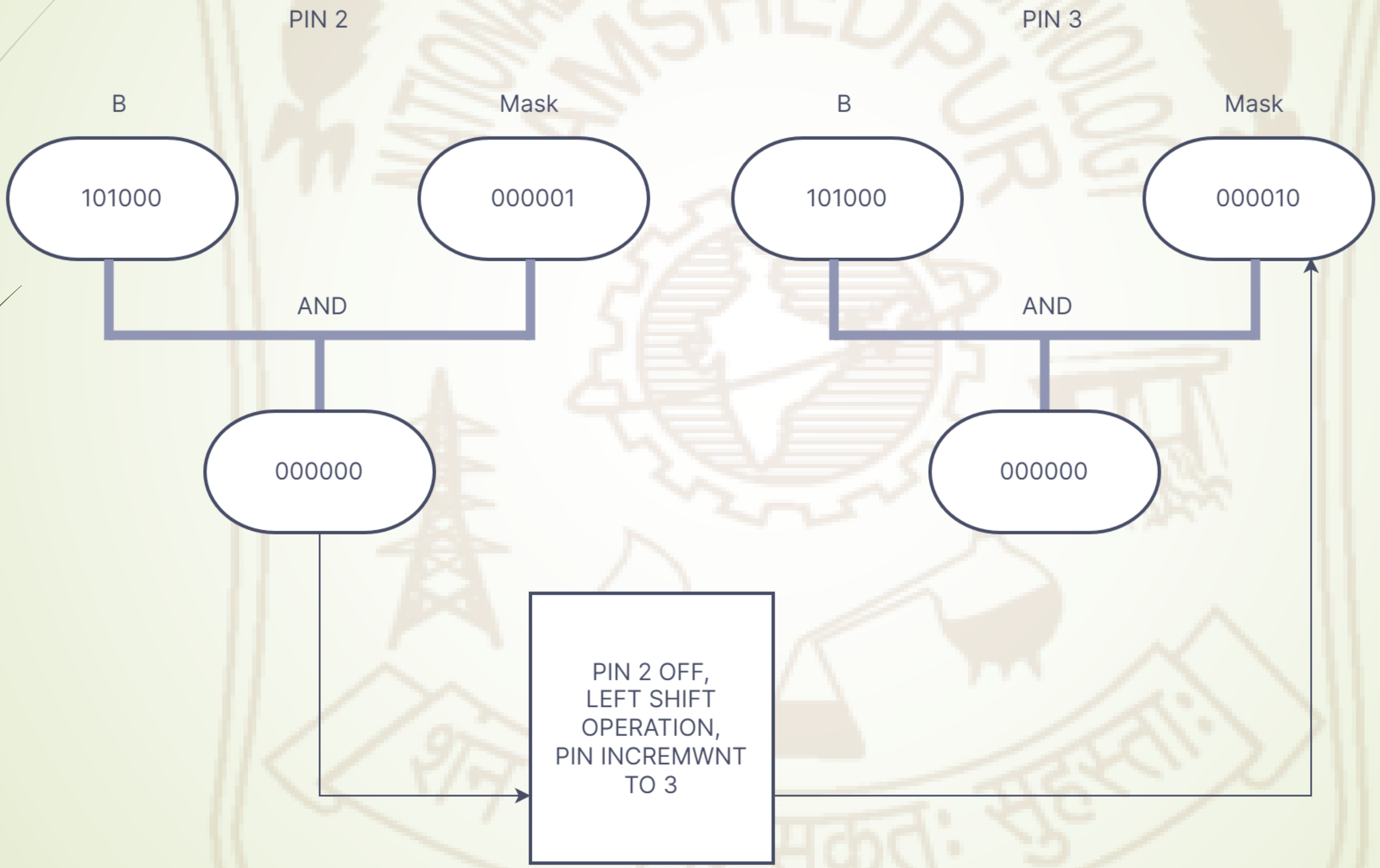


ASCII TABLE

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	[END 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]

Courtesy:Wikimedia.org

RESULTS & DISCUSSION

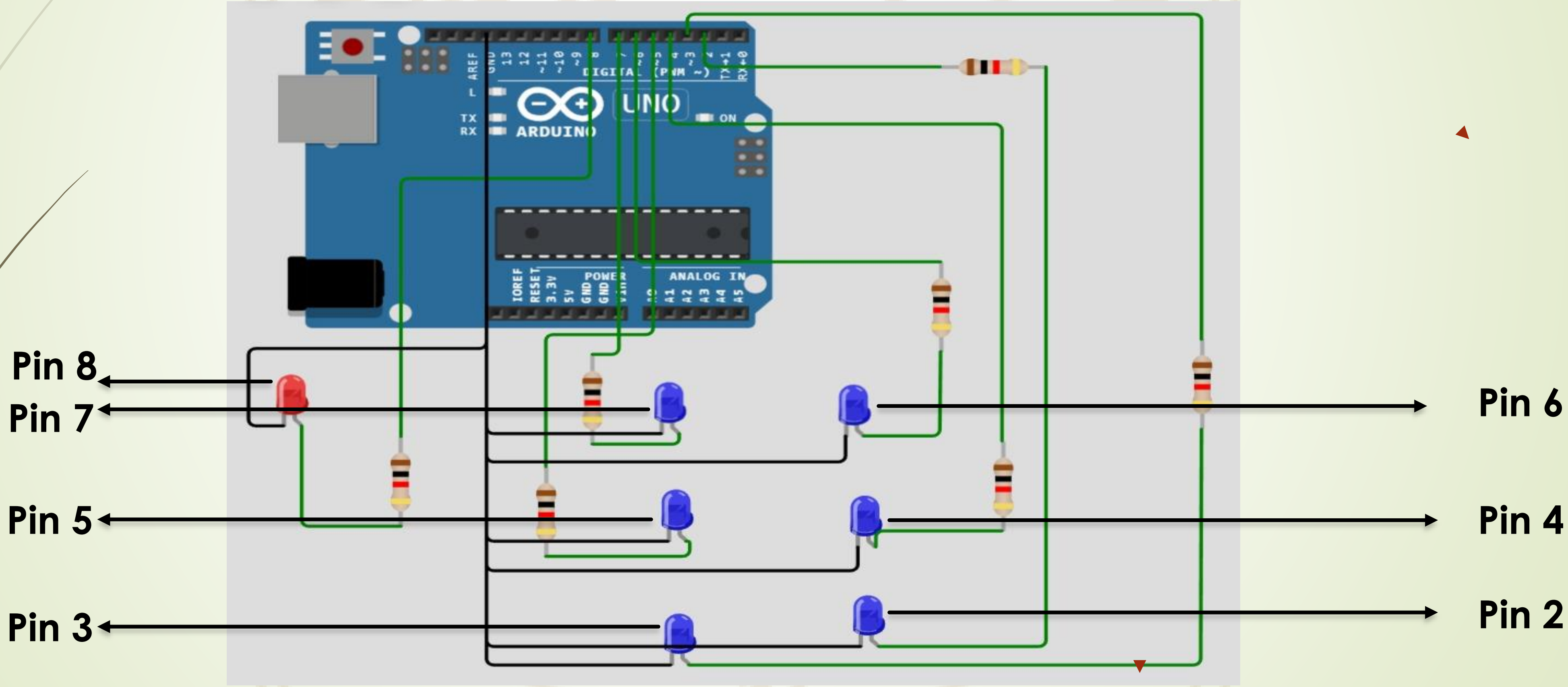


For Loop Flow Chart

Courtesy: Self-Made using Zen Flowchart

RESULTS & DISCUSSION

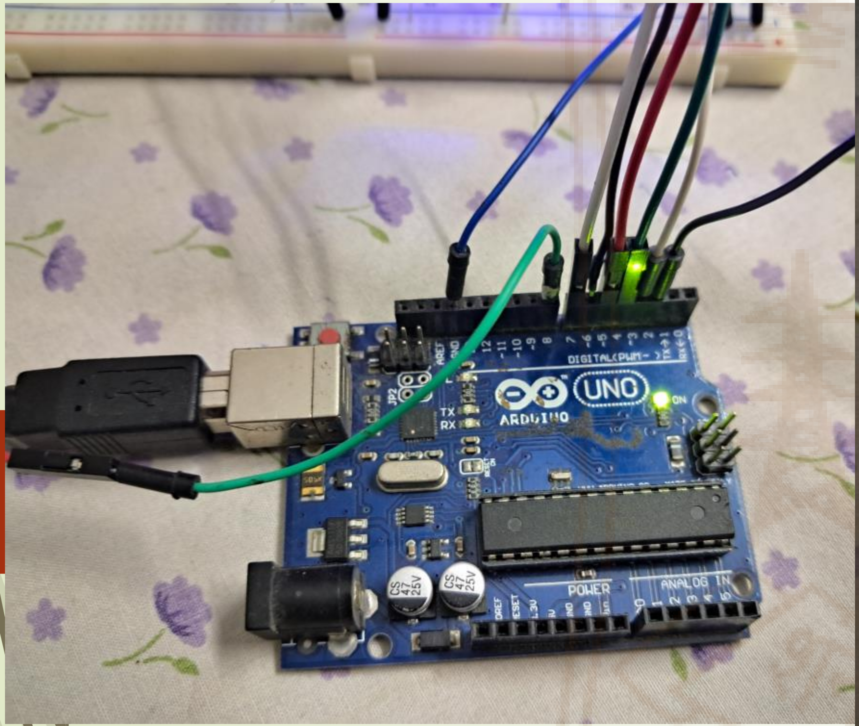
TEXT TO BRAILLE CODE IMPLEMENTATION:

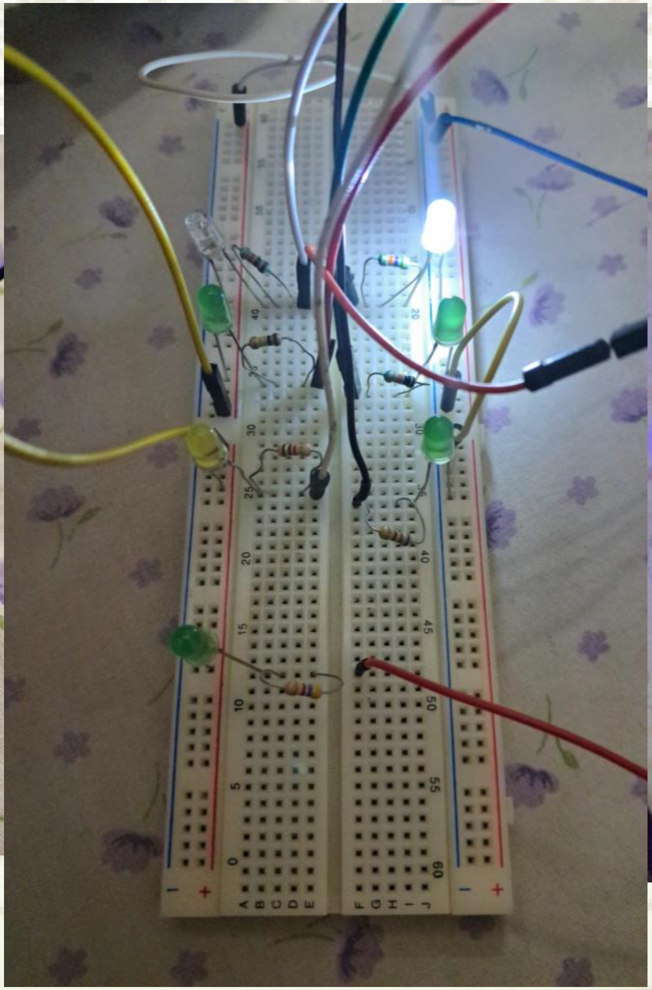


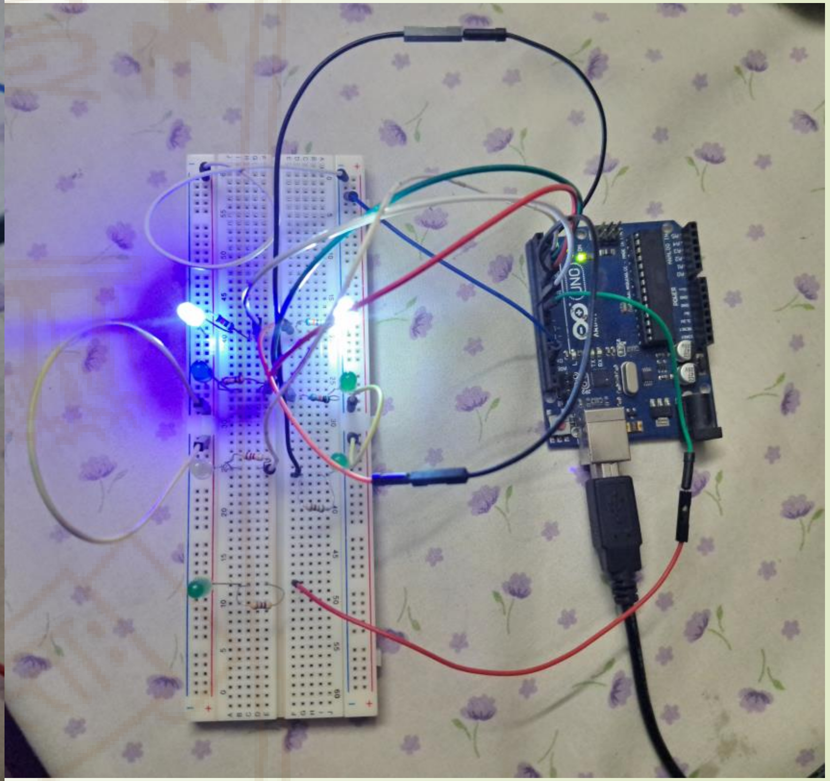
The implementation of the Text to Braille conversion on online simulator (www.wokwi.com)

RESULTS & DISCUSSION

HARDWARE:



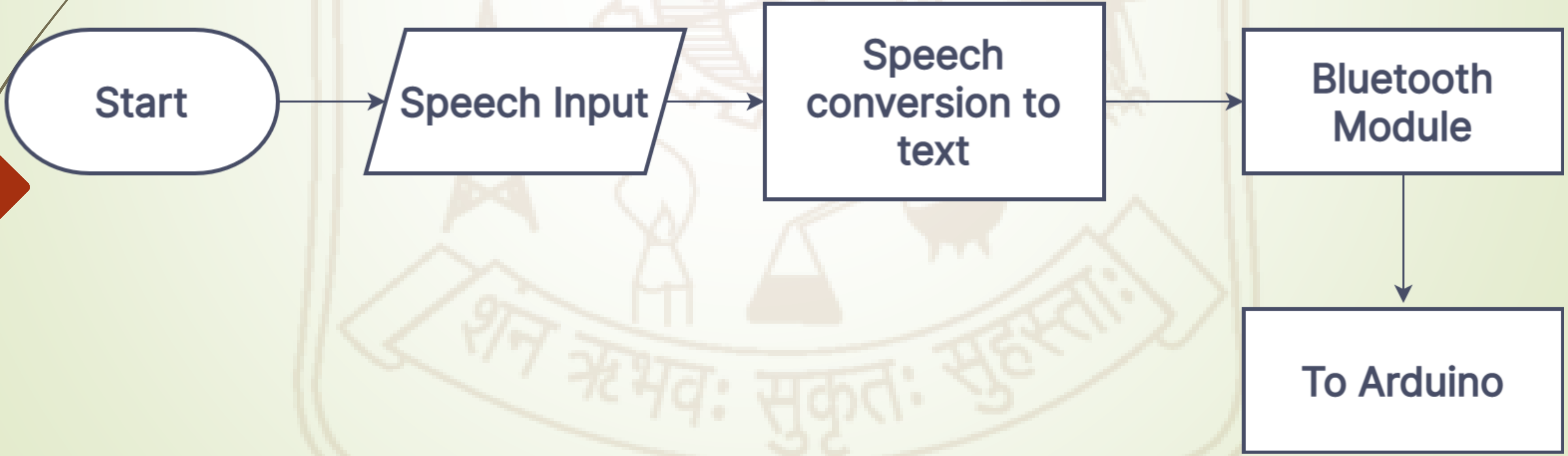




FUTURE WORK

2ND PHASE:

It will include speech recognition and text conversion. This will be achieved using a Bluetooth module which will be integrated with the basic braille translator.



FUTURE WORK

- Integrating Phase-I and Phase-II (i.e. speech to braille script conversion).

- Adding various languages including English, Hindi, etc and conversion to Braille.

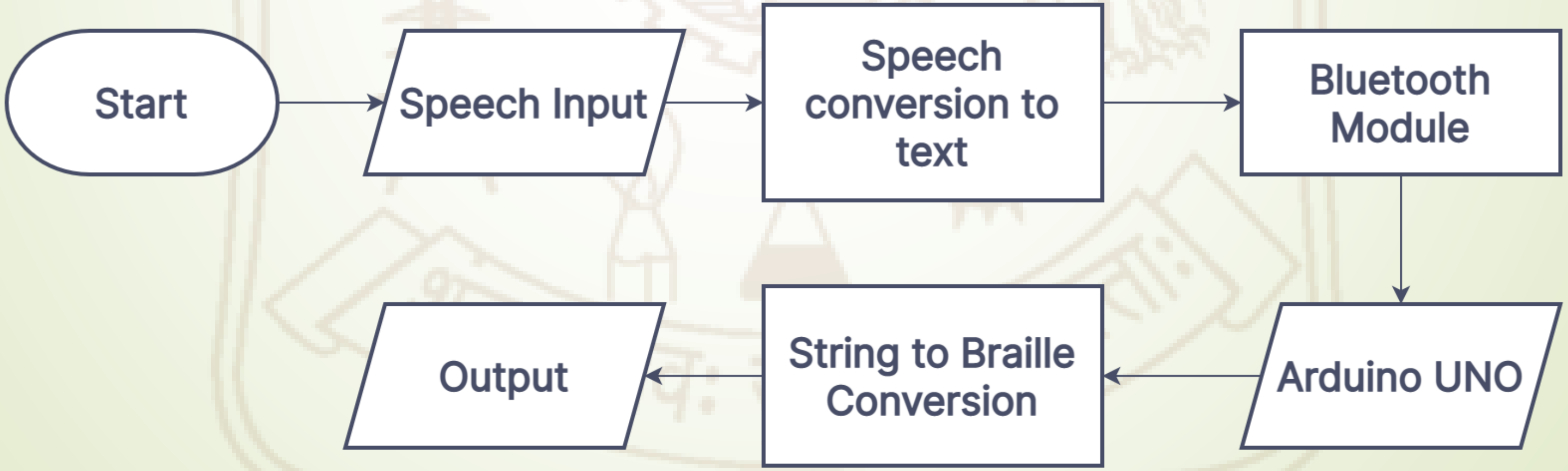
- Improving the cost-effectiveness of the whole setup, to increase the scalability.

- Making the setup user-friendly through constant improvements.

CONCLUSION

- The first phase of the project Multi-Lingual Speech to Braille Conversion has been successfully completed.

- The next phase includes the integration of the speech-to-text converter with the text-to-Braille converter using the Bluetooth module.



THANK

YOU