Assistive Device for the Speech Impaired

KAMALAKKANAN RAVI, PRAVEEN I.V., BALAMURALI M. AND POONGUZHALI S.

Speech-impaired people have difficulty in communicating with normal people because the hand gestures used by them to communicate their information is not easily understandable; only trained people can understand these. Most expressions and emotions remain un-conveyed, sometimes even misinterpreted. So hand gestures (limited gestures) are not an effective method of communication for the speech-impaired people.

To take care of this issue, an assistive device can be used to enhance the passage of communication. Here, the assistive device consists of a specialised keypad in which each key corresponds to a pre-assigned recorded audio that can be played to convey the message by the speech-impaired person.

Circuit and working

The device consists of transmitter and receiver sections. In the transmitter section (Fig. 1), the keypad matrix is connected to digital input/output (I/O) pins 2-7 of Board1 (Arduino Uno), which is used as the input device. The microcontroller (MCU) on Board1 scans the key being pressed and sends a 4-bit code representing the key pressed to I/O pins 10 to 13 of Arduino board.

These four bits are transferred to HT12E encoder for conversion of parallel data into serial data, which is fed into the RF transmitter. It then transmits this serial data at 433MHz using ASK modulation. The 750k resistor determines the oscillator frequency of the transmitter side.

In the receiver section, the code is received using the RF receiver and data bits are decoded if and only if the extracted address from the transmitted code matches the pre-programmed address in HT12D address pins, thereby avoiding interferences from the commercially-available receivers.

If the decoded address and pre-programmed address matches, LED1 glows, which is connected at pin 17 of IC HT12D. A 33-kilo-ohm resistor (R8) determines the oscillator frequency of the receiver. The decoded output from HT12D is sent to Board2 (Arduino Uno). Arduino board is programmed to generate a binary code, which is sent to IC 74HCT154 decoder to further trigger voice IC6 APR33A3.

In response to the trigger from the decoder, the voice IC generates corresponding audio messages programmed earlier.

The number of I/O pins on Arduino board is insufficient to drive voice IC APR33A3. So,

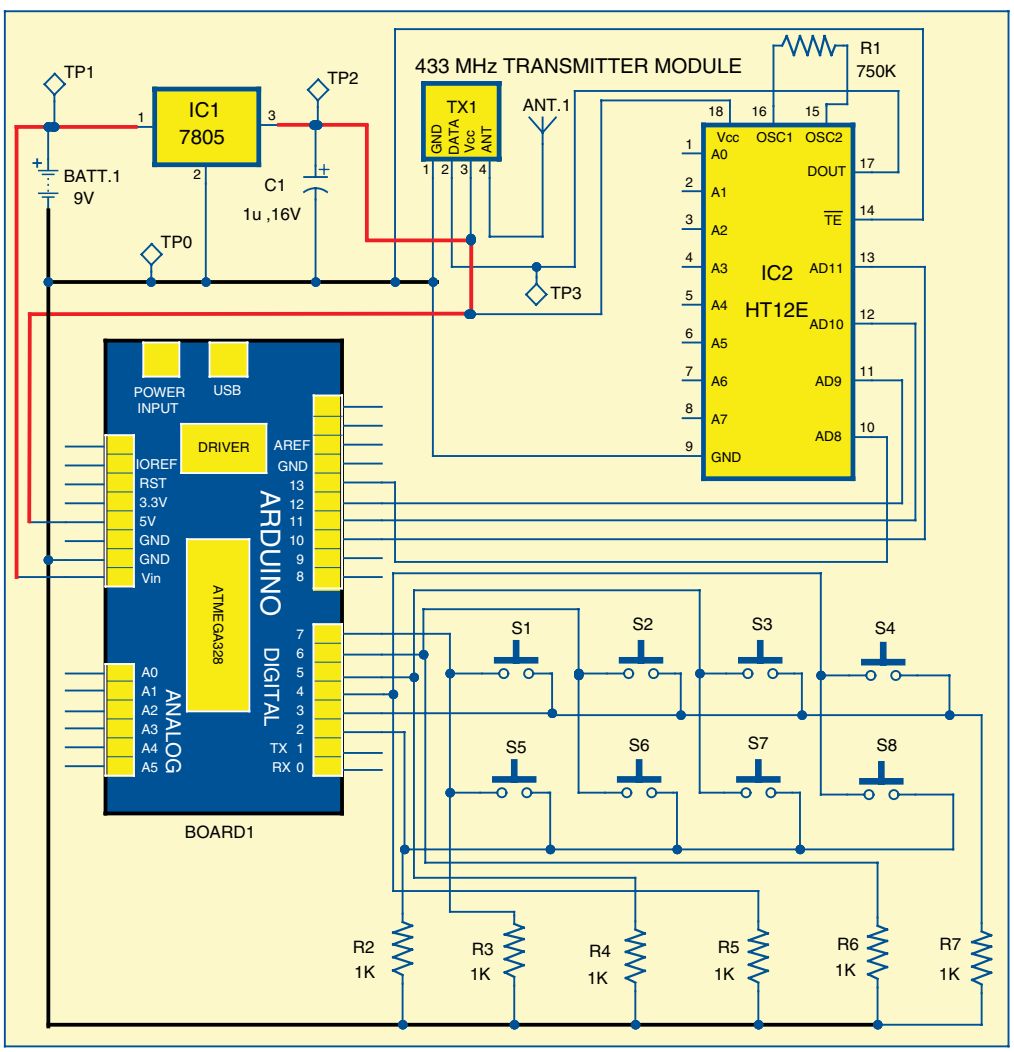

Fig. 1: Circuit of the transmitter section

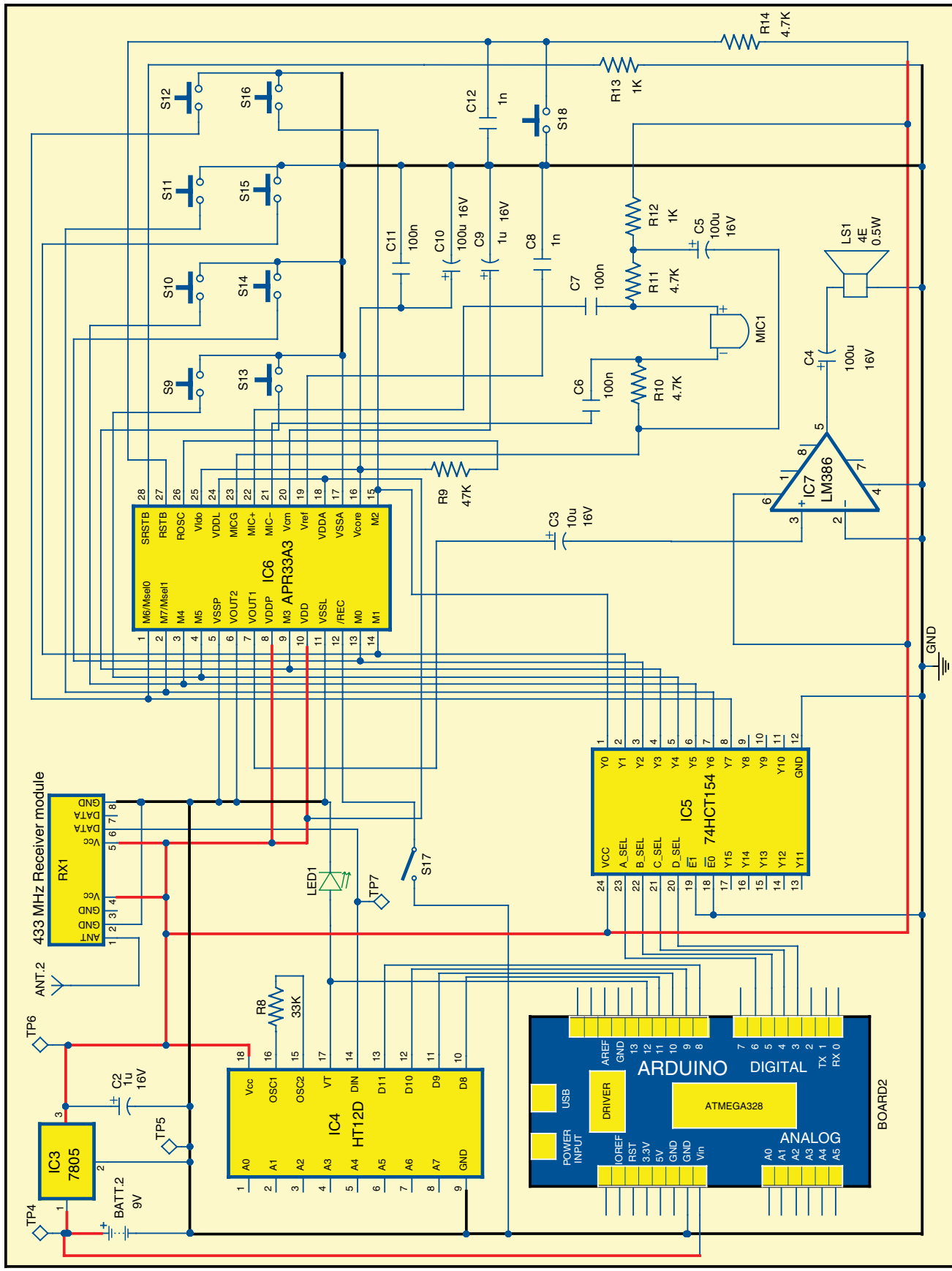

Fig. 2: Circuit of the receiver section

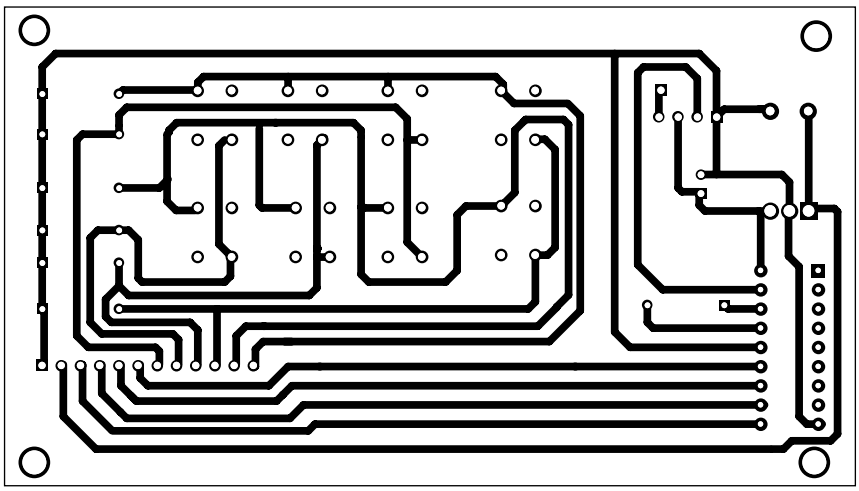

Fig. 3: Actual-size PCB pattern of the transmitter section

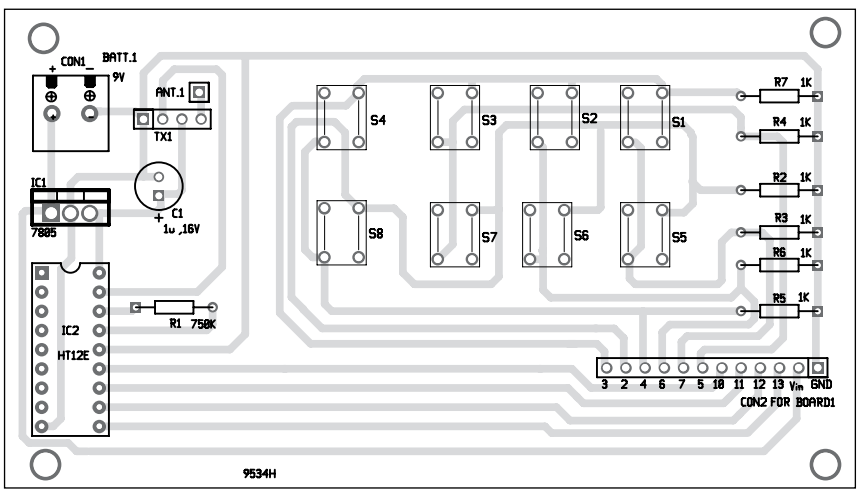

Fig. 4: Component layout of the transmitter section

74HCT154, a 4-to-16 decoder is used to drive the select pins of the voice IC (IC6).

For recording the voice, make REC pin of APR33A3 logic low by closing switch S17. Then, drive message switches (S9-S16) connected to APR33A3 to logic low, sequentially, as per the number of audio messages to be recorded. The chip will play back a beep and the message can then be recorded. When the pin is released, the chip will play back the beep twice to indicate that the message has been recorded.

To play back the message recorded, REC pin should be driven to logic high and then message switches (S9-S16) should be driven from logic high to logic low, and the message play back starts. (LM386 is a low-voltage audio power amplifier. The voltage gain ranges from 20 to 200.)

Construction and testing

An actual-size, single-side PCB of the transmitter section is shown in Fig. 3 and its component layout in Fig. 4. Similarly, an actual-size, single-side PCB for the receiver section is shown in Fig. 5 and its component layout in Fig. 6.

Assemble the circuit on the PCBs as it minimises time and assembly errors. Carefully assemble the components and double-check for any overlooked error(s). Use proper IC base for the MCU and other ICs.

Message switches at the receiver side and the number of switches at the transmitter side of the keypad may be increased as per requirement,

PARTS LIST

Semiconductors:

IC1, IC3	- 7805 voltage regulator
IC2	- HT12E encoder
IC4	- HT12D decoder
IC5	- 74HCT154, 4-to-16 line decoder
IC6	- APR33A3 voice record and play back IC
IC7	- LM386 low-voltage audio power amplifier
LED1	- 5mm LED
TX1	- 433MHz RF transmitter module
RX1	- 433MHz RF receiver module
BOARD1, BOARD2	- Arduino UNO board

Resistors (all 1/4-watt, ±5% carbon):

R1	- 750-kilo-ohm
R2-R7, R12, R13	- 1-kilo-ohm
R8	- 33-kilo-ohm
R9	- 47-kilo-ohm
R10, R11, R14	- 4.7-kilo-ohm

Capacitors:

C1, C2, C9	- 1µF, 16V electrolytic
C3	- 10µF, 16V electrolytic
C4, C5, C10	- 100µF, 16V electrolytic
C6, C7, C11	- 100nF ceramic
C8, C12	- 1nF ceramic

Miscellaneous:

CON1, CON3	- 2-pin connector
CON2	- 12-pin connector
CON4	- 5-pin connector
CON5	- 6-pin connector
LS1	- 4-ohms, 0.5W speaker
MIC1	- Electret condenser microphone
ANT.1, ANT.2	- 30cm-long single-strand wire antenna
BATT.1, BATT.2	- 9V battery
S1-S16, S18	- Tactile switch
S17	- On/off toggle switch
	- USB A-B cable

depending on the number of I/O pins available on Arduino UNO boards.

CON2 is a 12-pin connector used to interface the transmitter side PCB with Arduino UNO (Board1) as per the pins mentioned on the component side as shown in Fig. 4.

Similarly, CON4 is a 5-pin connector and CON5 is a 6 pin connector used to interface the receiver-side PCB with Arduino UNO (Board2) as per the pins mentioned on the component side as shown in Fig. 6.

CON1 and CON3 are 2-pin connectors used to connect 9V battery. To program the MCUs in Arduino boards, use a USB-A-B cable to connect to your system.

For troubleshooting, verify the voltages listed in test points table.

Software

The source code of the transmitter side contains a 2D array for the

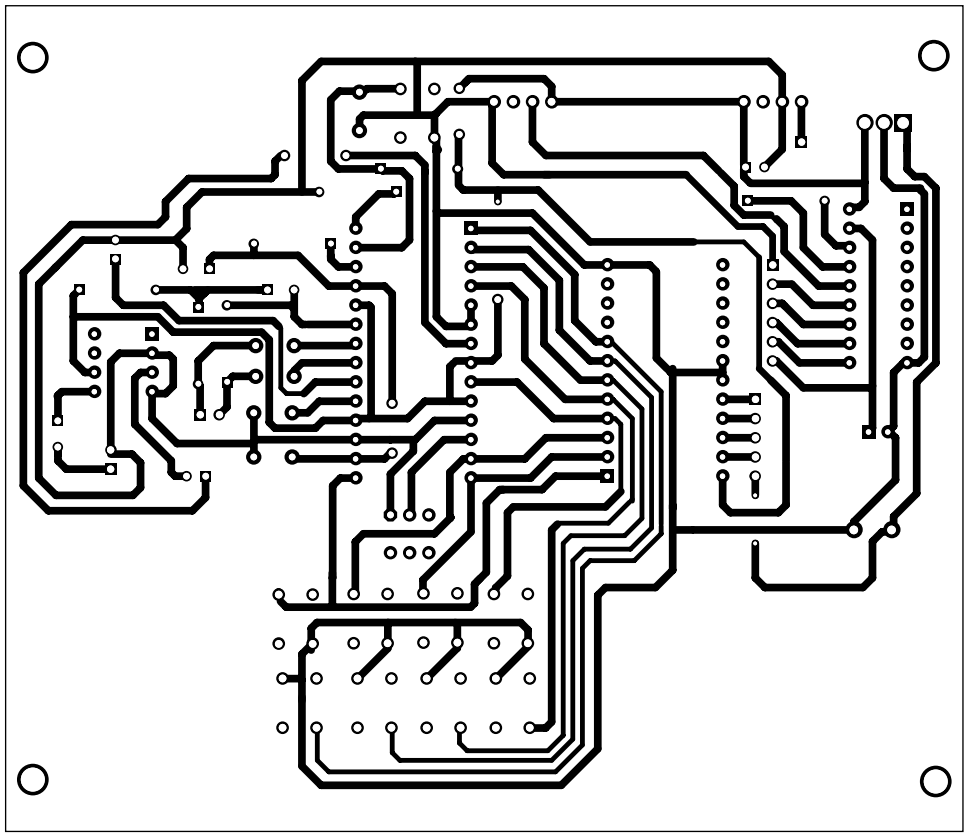

Fig. 5: Actual-size PCB pattern of the receiver section

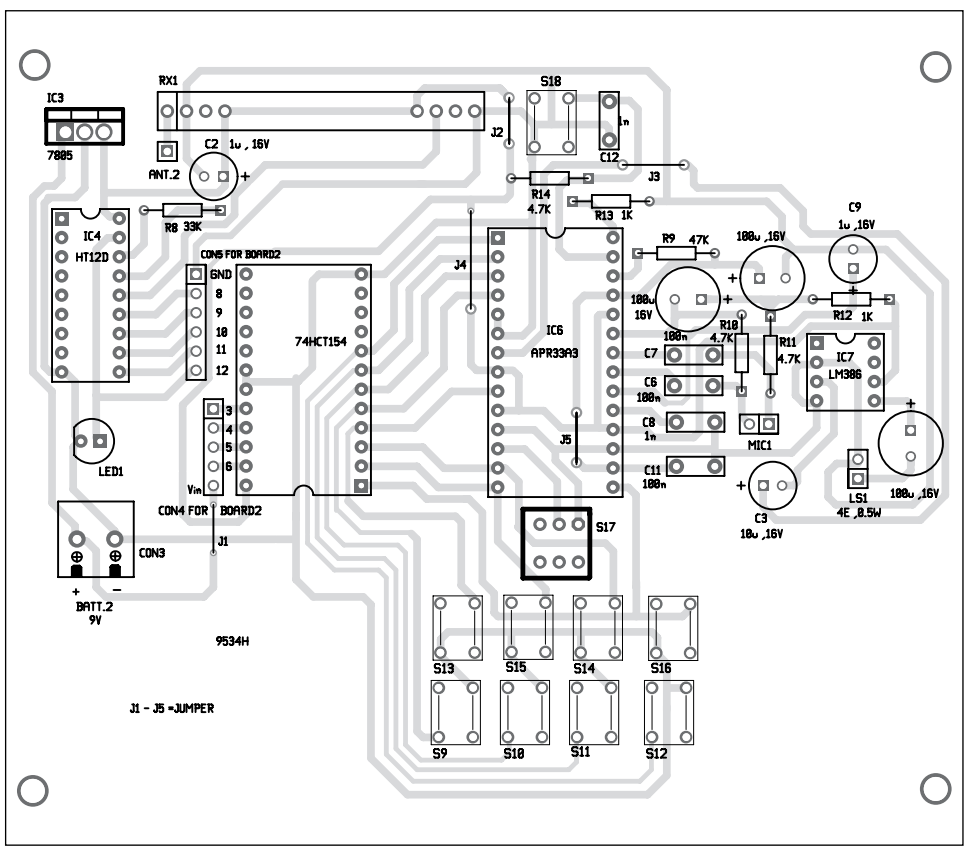

Fig. 6: Component layout of the receiver section

Test Points

Test point	Details
TP0, TP5	0V, GND
TP1, TP4	+9V DC
TP2, TP6	+5V DC
TP3	Serial data to TX1
TP7	Serial data from RX1

EFY Note

The source codes of this project are included in this month's EFY DVD and are also available for free download at *source.efymag.com*

pressed key. The code is modified to work for 16 switches for your convenience, so that if you want to increase the message switches and keypad switches apart from those mentioned in the PCB designs of the project, you will simply have to modify the hardware, and no changes would be required in the software section.

Initially, the code configures the column pins and row pins for the keypad and then in the setup ( ) function, it configures the output pins for Arduino UNO Board1.

Loop function is used to read the keypad and Digital Write ( ) function is used to output the data bits corresponding to read character.

In the receiver section code, I/O pins are configured for Arduino UNO Board2 and then the serial port is enabled for viewing the pressed key info on the serial monitor as well.

Loop ( ) function is used to read the receiving bits and convert these from binary to decimal values and send the received data to the output pins. ●

Poonguzhali S. is assistant professor at Department of Electronics and Communication, College of Engineering, Anna University, Guindy, Chennai. All other authors are biomedical engineering students at College of Engineering