Text-Generating Device (TGD)

Project Report

S.No	Name	Section	Enrollment Number
1	Shivendra Singh	S-5	U101116FCS117
2	Jatin Gupta	S-4	U101116FCS052
3	Shivank Pahwa	S-6	U101116FCS116
4	Shreyash Tiwari	S-6	U101116FCS244

ABSTRACT

Every day, people who are dumb (those who cannot speak), diagnosed with ALS (amyotrophic lateral sclerosis) or those who are diagnosed with severe physical impairment face problems while communicating with the general mass because of the lack of medium to express themselves.

Our project - **Text-Generating Device**, is built with the primary objective of giving out outputs in a **6x7 matrix** form containing 42 characters (all **26 letters** in the alphabet + commonly used **special characters**). The device comprises of horizontal and vertical alignment of LEDs, first running in the horizontal direction and then, in the vertical direction. Assisted by the push button, the user first stops the LED at the desired output in the horizontal direction followed by the vertical direction.

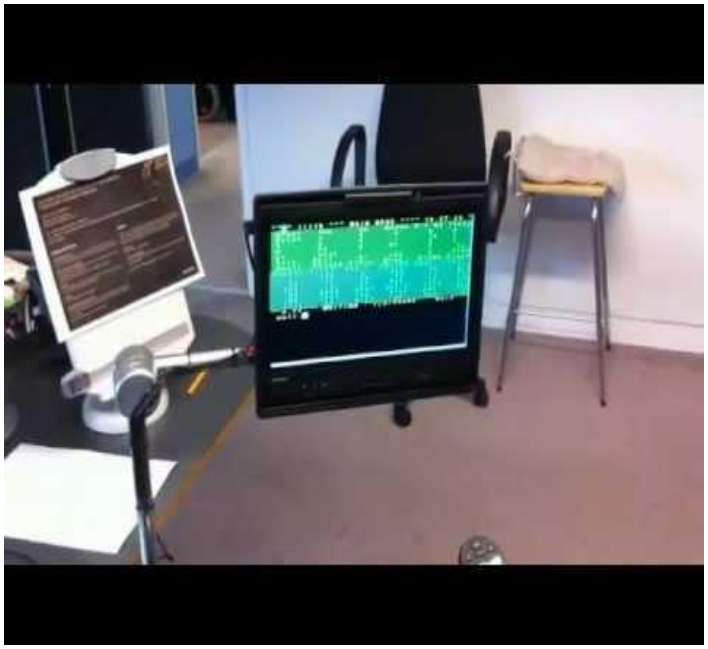
Hence, the user collectively chooses a combination of 2 pin numbers which would give a certain output on the display screen. The matrix on display is the following:

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U
V	W	X	Y	Z	1	2
3	4	5	6	7	8	9
SPACE	?	,	.	/	!	&

Inspiration for Text-Generating Device

Stephen Hawking is the former Lucasian Professor of Mathematics at the University of Cambridge and author of A Brief History of Time which was an international bestseller. Now the Dennis Stanton Avery and Sally Tsui Wong-Avery Director of Research at the Department of Applied Mathematics and Theoretical Physics and Founder of the Centre for Theoretical Cosmology at Cambridge, his other books for the general reader include A Briefer History of Time, the essay collection Black Holes and Baby Universe and The Universe in a Nutshell.

Stephen Hawking's Speech-Generating Device



What Hawking uses to communicate is aptly named a speech-generating device (SGD) or a voice output communication aid. This is a special device that either supplements or replaces speech/writing for people who have problems communicating in a traditional way.

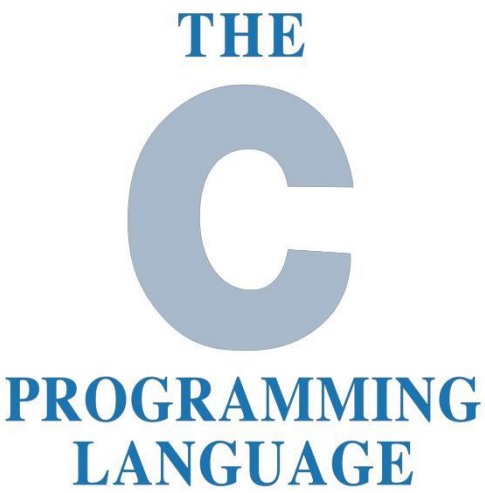
Hawking has been using this computer-based communication system made by Intel Corporation since 1997. The entire computer system is replaced every two years to cope with the gradual loss of control over his muscles over time.

Applications/Potential Users of Text-Generating Device

Built on the lines of the famous Speech-Generating Device produced by Intel Corporation for Stephen Hawking, the device aims to garner people with certain disabilities:

- Amyotrophic Lateral Sclerosis (ALS)
- Dumb
- Other disabilities involving severe physical weakness combined with dumbness

Language used: C Programming Language



C is an imperative procedural language, designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently, and to require minimal run-time support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code. The language has become available on a wide range of platforms, from embedded microcontrollers to supercomputers.

Arduino Libraries (Used in C)

Libraries are files written in C or C++ (.c, .cpp) which provide your sketches with extra functionality (e.g. the ability to control an LED matrix, or read an encoder, etc.). They were introduced in Arduino 0004.

How: To use an existing library in a sketch simply go to the Sketch menu, choose "Import Library", and pick from the libraries available. This will insert an #include statement at the top of the sketch for each header (.h) file in the library's folder. These statements make the public functions and constants defined by the library available to your sketch. They also signal the Arduino environment to link that library's code with your sketch when it is compiled or uploaded.

User-created libraries as of version 0017 go in a subdirectory of your default sketch directory. For example, on OSX, the new directory would be ~/Documents/Arduino/libraries/. On Windows, it would be My Documents\Arduino\libraries\. To add your own library, create a new directory in the libraries directory with the name of your library. The folder should contain a C or C++ file with your code and a header file with your function and variable declarations. It will then appear in the Sketch | Import Library menu in the Arduino IDE.

Hardware

1. **Push Button:** A push button switch is a small, sealed mechanism that completes an electric circuit when you press on it. When it's on, a small metal spring inside makes contact with two wires, allowing electricity to flow. When it's off, the spring retracts, contact is interrupted, and current won't flow.

2. **Arduino Mega 2560:** The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

3. **Jumper Wire:** A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable– named for one manufacturer of them) is an electric wire or group of them in a cable with a connector or pin at each end (or sometimes without them– simply "tinned"), which is normally used to interconnect the components of a breadboardor other prototype or test circuit, internally or with other equipment or components, without soldering.

4. **Multi Purpose PCB:** A printed circuit board(PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conducive substrate. Components (e.g. capacitors, resistor or active devices) are generally soldered on the PCB. The PCB used in our project is of dimesion 15*10 cms.

5. **Bread Board:** A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. It has strips of metal underneath the board and connect the holes on the top of the board.

6. **Arduino Cable:** An Arduino cable is a standard issue USB 2.0 Cable. This is the most common A to B Male/Male type peripheral cable. It is mostly compatible with SFE designed USB boards as well as various Arduino boards like Mega 2560, Uno, etc.

7. **LED Display:** An LED display, or light emitting diode display, is a flat panel display that uses light emitting diodes as the video display. LED diodes are used in order to make up an LED display. A light emitting diode is made up of a semiconductor chip which is surrounded by a transparent plastic case. The plastic case allows the light to pass through it.

8. **Solder Wire:** Solder is a fusible metal alloy used to create a permanent bond between metal workpieces. Solder must be melted in order to adhere to and connect the pieces together, so a suitable alloy for use as solder will have a lower melting point than the pieces it is intended to join.

9. **Resistor 390:** Resistor 390 is a 390 ohm/ 0.5 watt resistor. In the project, all the LED's have made use of resistor 390.

Resistor 470: Resistor 470 is a 470 ohm/ 0.25 watt resistor. It has 5% tolerance with the following colour code: Yellow, Violet, Brown, Gold. Resistor 470 has been used in LED display in the project.

Resistor 5.6K: Resistor 5.6K is a 5600 ohm/ 0.5 watt resistor. It has 5% tolerance with following
colour code: Green, Blue, Red and Gold. Resistor 5.6K has been used in the push button.

Code used in the Device

```
#include <Wire.h>
#include <LiquidCrystal.h>
int inpbutton;
int inpbutton2;
int i, j,k,l,m,o,p;
int topmin=26;
int sidemin=29;
int f_l=1;
int f_2=1;
char c;
const int rs =8 , en = 9, d4 =4 , d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
 lcd.begin(16, 2);
 pinMode(53,INPUT);
  for(int i=22;i<41;i++)
  {
    pinMode(i, OUTPUT);
  }
  Serial.begin(9600);


}
char DispLCD(int top, int side){
  char disp[7][7]={{'A','B','C','D','E','F','G'},
                   {'H','I','J','K','L','M','N'},
                   {'O','P','Q','R','S','T','U'},
                   {'V','W','X','Y','Z','1','2'},
                   {'3','4','5','6','7','8','9'},
                   {' ','?',',',' ','.','/','!','&'}};
  return disp[top][side];
}
int Top(){
```

```
 for(i=26;;i+=2)
  {
   digitalWrite(i, HIGH);   // turn the LED on (HIGH is the voltage level)
   delay(500); // wait for a second
   digitalWrite(i, LOW);    // turn the LED off by making the voltage LOW
   delay(500);
   inpbutton = digitalRead(53);
   Serial.println(inpbutton);
   if(inpbutton==f_l){
    return i;
   }
   if(i==38){
    i=24;
   }

    }
   inpbutton=0;
   }
int Side(int lck){
  digitalWrite(lck, HIGH);
 for(i=29;;i+=2)
  {
   digitalWrite(i, HIGH);   // turn the LED on (HIGH is the voltage level)
   delay(500); // wait for a second
   digitalWrite(i, LOW);    // turn the LED off by making the voltage LOW
   delay(500);
   inpbutton = digitalRead(53);
   Serial.println(inpbutton);
   if(inpbutton==f_l){
    return i;
   }
   if(i==39){
    i=27;
   }

    }
   inpbutton=0;
   }


void loop() {
```

```
   inpbutton=0;
   inpbutton = digitalRead(53);

   inpbutton2=0;
   inpbutton = digitalRead(52);
   if(inpbutton2=f_2){
     lcd.clear();
   }

   j=Top();

   k=Side(j);
  digitalWrite(j, HIGH);
  digitalWrite(k, HIGH);
  delay(3000);
  digitalWrite(j, LOW);
  digitalWrite(k, LOW);
  l=-1*(topmin-j);

  m=-1*(sidemin-k);
  o=l/2;
  p=m/2;
  c=DispLCD(p,o);
  lcd.print(c);
  inpbutton=0;


}
```