CODE:

Arduino Code:

```
String data;

String data1;


void setup() {

  Serial.begin(9600); // voice in raspberry pi

  Serial1.begin(9600); // GSM receive SMS

  Serial2.begin(9600);  //LED display


  Serial1.print("AT+CMGF=1\r");  // set SMS mode to text

  delay(100);

  Serial1.print("AT+CNMI=2,2,0,0,0\r");

  // blurt out contents of new SMS upon receipt to the Serial1 shield's serial out

  delay(100);

  pinMode(13,OUTPUT);

  Reset();

}

//////////////////////////////////////////////////////


 String processSerialData(String SerialData){

  SerialData = SerialData.substring(1);

  int crlnd = SerialData.indexOf(0x0D);

  String actualData = SerialData.substring(crlnd+1,SerialData.lastIndexOf(0x0D));
```

```
  if(actualData.charAt(0)==0x0A){
     actualData = actualData.substring(1);
    }
  if(actualData.charAt(actualData.length()-1)==0x0A){
     actualData = actualData.substring(0,actualData.length()-1);
    }
   return actualData;
  }
////////////////////////////////////////////////////////////////
void Reset(){
  for(int i=0;i<2;i++){
    delay(2000);
    Serial2.println("#");
   }
  delay(2000);
  Serial2.println("4");
  delay(1000);
  Serial2.println("1");
  delay(1000);
}

////////////////////////////////////////////////////////////////
void loop() {
if(Serial1.available() >0){
```

```
   digitalWrite(13,HIGH);
   String incoming = Serial1.readString();
   String data = processSerialData(incoming);
   Reset();
   Serial.print(data);
   String data1 = data;
   delay(1000);
     for(int i=0;i<2;i++){
       Serial2.println("<M "+data+"><S 5><D L1>");
       delay(500);
       Serial2.println("6");
       delay(100);
      }
     }
  else{
   digitalWrite(13,LOW);
   Serial.print(data1);
   delay(3000);
    }
  }
```

Raspberry Pi Code:

```
#!/usr/bin/env python3
import serial
```

```
import os

import time


os.system("ls -l")

##port = serial.Serial('/dev/ttyUSB0',9600,timeout=1)


from subprocess import call



cmd_beg= 'espeak '

cmd_end= ' | aplay /home/pi/Desktop/Text.wav  2>/dev/null' # To play back the stored .wav file and to dump the std errors to /dev/null

cmd_out= '--stdout > /home/pi/Desktop/Text.wav ' # To store the voice file

while True:

    #text = port.read(99) ## read serial data and store.

    text = input("enter the text : ")

    print(text)

    voi = str(text)

    print(voi)


    #Replacing ' ' with '_' to identify words in the text entered

    voi = voi.replace(' ', '_')

    #voi = voi[1:]

    for x in range(3):

        #Calls the Espeak TTS Engine to read aloud a Text
```

```
call([cmd_beg+cmd_out+voi+cmd_end], shell=True)

os.system("aplay /home/pi/Desktop/Text.wav")

time.sleep(1)
```