```
/*Download all libraries from GitHub.
* note*--The message send ove bluetooth is cleared if the Arduino is shut down (power off), and when
   you first
* Start the Arduino the the characters written into the sketch will be display.
 */

#include <SPI.h>        //Include all of these libraries to avoid compiling error
#include <DMD.h>
#include <TimerOne.h>
#include "SystemFont5x7.h"
#include "Arial_black_16.h"

#define DISPLAYS_ACROSS 2  //Mention how many DMD you're using(In my case 2)
#define DISPLAYS_DOWN 1   //DMD up/down
DMD dmd(DISPLAYS_ACROSS, DISPLAYS_DOWN);
#define max_char 1000 //Define Nos. of characters you're using  char message[max_char];
char r_char;
byte index = 0;
int i;
char greeting[] = "Subscribe to IndianDIYers"; //Print your message max. 1000 characrer

 void ScanDMD()
{
 dmd.scanDisplayBySPI();
}

 void setup(void)
{
 Timer1.initialize( 5000 );
 Timer1.attachInterrupt( ScanDMD );     dmd.clearScreen(
true );//to clear RAM    Serial.begin(9600);//begin serial
communication   strcpy(message,greeting);
 }
void loop(void)
```

```
  {
  if(Serial.available())
  {
  for(i=0; i<999; i++){
message[i] = '\0';
       }

index=0;
    }
  while(Serial.available() > 0){

  dmd.clearScreen( true );
if(index < (max_char-1))
      {
  r_char = Serial.read();
message[index] = r_char;
index++;



 }
}

 dmd.selectFont(Arial_Black_16);

 dmd.drawMarquee(message ,max_char,(32*DISPLAYS_ACROSS)-1
,0);  long start=millis();  long timer=start;  boolean ret=false;
while(!ret)
 {
 if ((timer+30) < millis()) {
ret=dmd.stepMarquee(-1,0);
timer=millis();
    }
   }
}
```