const int dotduriation = 50;

const int dashduriation = 150;

const int intercharspace = 50;

const int letterspace = 150;

const int wordspace = 300;

void dot(){

  digitalWrite(LED\_BUILTIN, HIGH);

  delay(dotduriation);

  digitalWrite(LED\_BUILTIN, LOW);

  delay(intercharspace);

}

void dash(){

  digitalWrite(LED\_BUILTIN, HIGH);

  delay(dashduriation);

  digitalWrite(LED\_BUILTIN, LOW);

  delay(intercharspace);

}

void morsecode(char letter){

    switch (letter) {

    case 'A': dot(); dash(); break;

    case 'B': dash(); dot(); dot(); dot(); break;

    case 'C': dash(); dot(); dash(); dot(); break;

    case 'D': dash(); dot(); dot(); break;

    case 'E': dot(); break;

    case 'F': dot(); dot(); dash(); dot(); break;

    case 'G': dash(); dash(); dot(); break;

    case 'H': dot(); dot(); dot(); dot(); break;

    case 'I': dot(); dot(); break;

    case 'J': dot(); dash(); dash(); dash(); break;

    case 'K': dash(); dot(); dash(); break;

    case 'L': dot(); dash(); dot(); dot(); break;

    case 'M': dash(); dash(); break;

    case 'N': dash(); dot(); break;

    case 'O': dash(); dash(); dash(); break;

    case 'P': dot(); dash(); dash(); dot(); break;

    case 'Q': dash(); dash(); dot(); dash(); break;

    case 'R': dot(); dash(); dot(); break;

    case 'S': dot(); dot(); dot(); break;

    case 'T': dash(); break;

    case 'U': dot(); dot(); dash(); break;

    case 'V': dot(); dot(); dot(); dash(); break;

    case 'W': dot(); dash(); dash(); break;

    case 'X': dash(); dot(); dot(); dash(); break;

    case 'Y': dash(); dot(); dash(); dash(); break;

    case 'Z': dash(); dash(); dot(); dot(); break;

    default: break;

  }

  delay(letterspace);

}

const char\* message = "my name is chinmayee";

void setup(){

  pinMode(LED\_BUILTIN, OUTPUT);

  digitalWrite(LED\_BUILTIN, LOW);

}

struct{

  char bytes[512];

  int idx;

} inbuf;

void loop(){

  if(port.available()){

    port.write(port.read());

  }

  for(int i=0; message[i] != '\0'; i++){

    if(message[i] == ' '){

      delay(wordspace);

    }

    else{

      morsecode(toupper(message[i]));

    }

  }

  delay(3000);

}