**Arduino: 간단한 터미널 프롬프트 구현**

Arduino에 간단한 터미널을 만들어 명령어를 통해 제어가 가능한 터미널을 만들어 보자.

다음 글은 Arduino에 간단한 터미널 알림부호를 구축하는 방법을 설명하겠습니다.

IDE를 사용하여 Arduino 프로그램을 작성하고 있습니다. 모든 코드 예시는 C++로 작성됩니다.  
  
터미널 프롬프트 베이스  
1. Arduino가 시작되면 작은 시작 메시지를 표시한다..

2. help를 치면 명령어를 어떻게 조작하는지에 대한 정보를 제공해야 한다.

3. 내가 어떤 명령을 입력할 때, 혹은 더 구체적으로 명령에 대한 설명이 제공되어야 한다.

4. 프롬프트를 시작하려면 다음 방법을 사용합니다.

void helpMsg() {

Serial.println("\

Please select a command\n\

- LED {msg} Print msg to LED Matrix \n\

- LOG {msg} Log msg \n\

- EXIT Restart the program\n");

}

문자열 문자로 표시하고 일련 번호로 인쇄할 여러 줄 문자열을 정의합니다.

다음은 사용자가 입력해야 한다는 힌트입니다.원하는 입력 알림 기호를 사용하십시오.

void promptMsg() {

Serial.print("> ");

}

setup() 기간에 이 두 가지 방법을 호출한다.

void setup() {

Serial.begin(9600);

helpMsg();

promptMsg();

}

void loop() { /\* ... \*/ }

**읽기 입력**  
Arduino 라이브러리는 입력을 읽는 많은 방법을 제공합니다. 가장 흔히 볼 수 있는 것은 Serial.read() 한 바이트의 속도로 직렬 버퍼를 보내고 char 로 되돌아오는 것이다.

즉각적인 피드백의 요구를 만족시키기 위해 우리는 어떠한 방법의 delay()도 사용할 수 없다. 그러나 이와 동시에, 우리는 명령이 완성되기를 기다려야 하고, 사용자가 리턴 버튼을 누르기를 기다려야 한다.  
해결 방안은 입력을 사용할 수 있을 때 즉시 읽고, 이 입력을 버퍼에 추가하고, 직렬 데이터로 출력하는 것이다.

String cmd;

char car;

void loop() {

if (Serial.available()) {

car = Serial.read();

cmd += String(car);

Serial.print(car);

}

if (cmd.indexOf('\n') > -1) {

process();

cmd = "";

promptMsg();

}

}

이 줄에서 우리는 Serial.read()를 사용하여 사용자의 입력을 읽고 값을 Char 변수에 저장합니다.이 문자는 Stringcmd에 추가되어 화면에 인쇄됩니다.  
다음에 cmd 문자열에 새 줄 문자가 포함되어 있는지 확인합니다. 이것은 사용자가 enter 키를 눌렀음을 의미합니다. 만약 그렇다면, 우리는 방법 process() 을 호출해서 다음 절을 참고하고 리셋 cmd 하고 다시 알림을 표시합니다.

**명령 처리**  
Enter 키를 누르면 함수process()가 호출됩니다. 그것은 입력한 데이터 cmd 를 조사하고 무엇을 할지 결정할 것이다. 마찬가지로 우리는 String류의 방법을 사용한다..indexOf()를 사용하면 문자열에 다른 하위 문자열이 포함되어 있는지 확인할 수 있습니다..substring를 사용하면 문자열에서 부분을 추출할 수 있습니다.  
다음 행에서는 cmd에 키워드LED, LOG 또는 EXIT가 포함되어 있는지 확인합니다.LED와 LOG에 대해 우리는 명령 후 유효한 하중이 있는지도 검사한다.일치하는 경우, 색인 cmd 의 [4, cmd.length() -2] 문자열에서 유효 부하를 추출하여 진일보한 처리를 진행할 것입니다.프레젠테이션 목적으로, 여기에서 우리는 일치하는 것만 인쇄합니다.

void process() {

int index;

if((index = cmd.indexOf("LOG")) > -1 && cmd.length() > 4) {

String msg = cmd.substring(index + 4, cmd.length() -2);

Serial.println("LOG >>" + msg + "<<");

} else if((index = cmd.indexOf("LED")) > -1 && cmd.length() > 4) {

String msg = cmd.substring(index + 4, cmd.length() -2 );

Serial.println("LED >>" + msg + "<<");

printStringOnMatrix(msg);

} else if((index = cmd.indexOf("EXIT")) > -1) {

Serial.println("Reboot ...");

}

}

결론  
본고는 Arduino에서 어떻게 간단한 터미널 프로그램을 실현하는지 보여 준다. Arduino를 Seral 모니터에 연결해서 명령 목록을 보고 상호작용을 시작할 수 있습니다.프로젝트에 LED 점진을 추가했습니다. LED 접두사가 있는 입력 텍스트는 모두 인쇄됩니다.