



Nazwa implementacji: Zapis danych na kartę pamięci microSD - kod do Arduino

Autor: Łukasz Ciężki

Opis implementacji: Implementacja precyzuje możliwość wykorzystania wcześniej zmontowanego układu do zapisu danych na kartę pamięci microSD za pomocą Arduino

Jeżeli układ został zmontowany na Arduino, to powinniśmy się teraz wziąć za kod. Do obsługi systemu plików karty pamięci potrzebne są odpowiednie biblioteki - dostępne są w załącznikach implementacji. Folder zawarty w archiwum należy wypakować do katalogu "libraries" w katalogu Arduino. Po skopiowaniu bibliotek możemy użyć naszego kodu:

```
#include <avr/io.h>
//biblioteki FatFS stworzone przez ChaN na licencji GNU GPL --- elm-chan.org
#include <diskio.h>
#include <ff.h>
#include <ffconf.h>
#include <integer.h>

FATFS FatFs;
FIL Fil;

DWORD get_fattime (){
//zwraca czas utworzenia/zmiany pliku
return ((DWORD)(2014 - 1980) << 25) //rok: 2014
| ((DWORD)5 << 21) //miesiąc: Maj
| ((DWORD)28 << 16) //dzień: 4
| ((DWORD)0 << 11) //godzina: 15
| ((DWORD)0 << 5) //minuta: 0
| ((DWORD)0 >> 1); //sekunda: 0
}

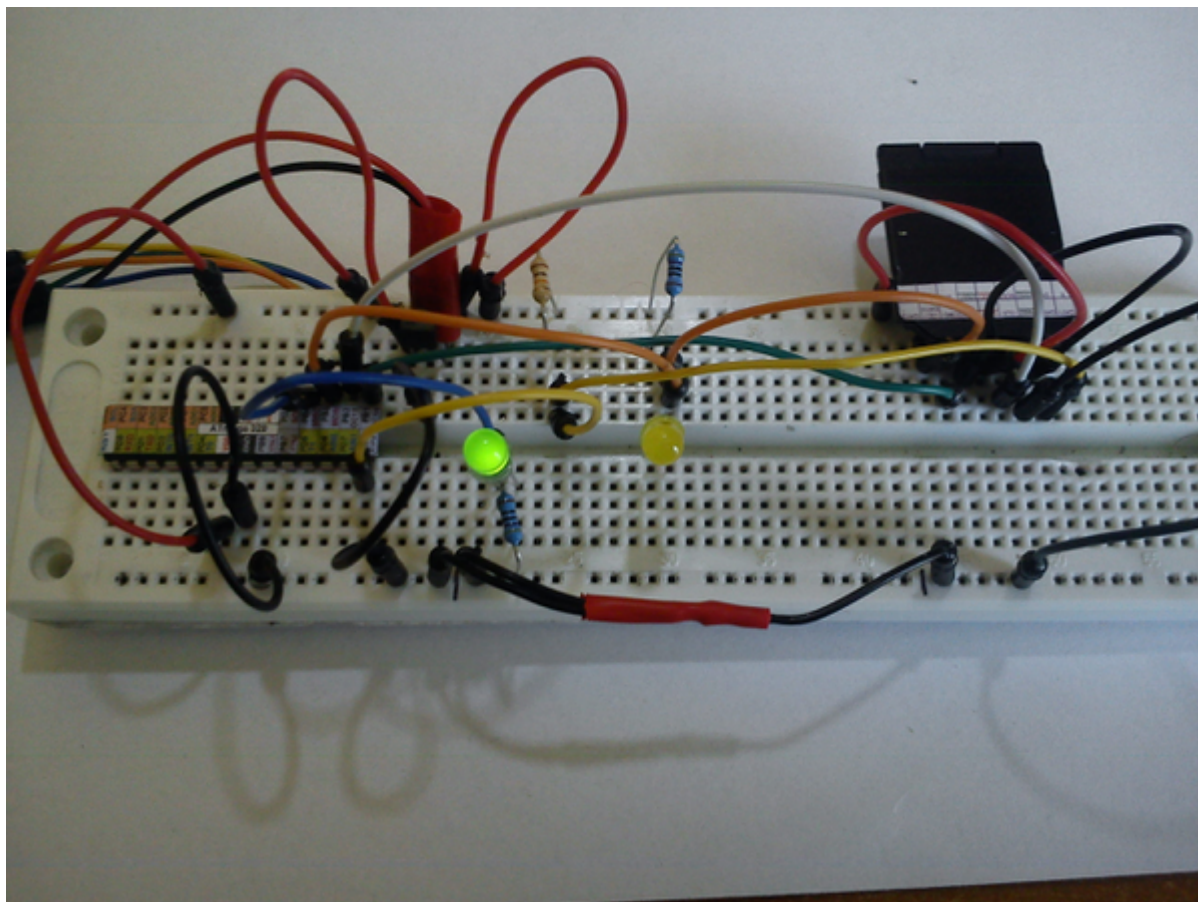
void setup(){
pinMode(12, OUTPUT);
}
void loop(){
unsigned int bw; //stwórz zmienną do przechowywania ilości znaków

f_mount(&FatFs, "", 0); //zamontuj dysk

if (f_open(&Fil, "naszplik.txt", FA_WRITE | FA_CREATE_ALWAYS) == FR_OK) { //jesli plik "naszplik.txt" nie istnieje na karcie pamięci, to stwórz go
f_write(&Fil, "Zapis sie powiodl\r\n", 17, &bw); //pisz do pliku 17 znaków
f_close(&Fil); //zamknij plik
if (bw == 17) { //jesli w pliku jest 17 znaków (jeśli poprawnie się zapisało)
digitalWrite(12, HIGH); //świeć diodą na 12 pinie
}
}
for (;;) ; //nic nie rob (nieskonczona pusta petla)
}
```

Komentarze wyjaśniają, do czego służą poszczególne operacje. Podczas pracy z kartą pamięci żółta (na zdjęciu) dioda będzie migać. Jeśli plik zapisze się poprawnie, zielona (na zdjęciu) dioda będzie się stale świecić.





Efekt naszej pracy

otwarty na komputerze powinien wyglądać mniej więcej tak:

