

Redundancja danych



cansats in europe
2019 polish competition



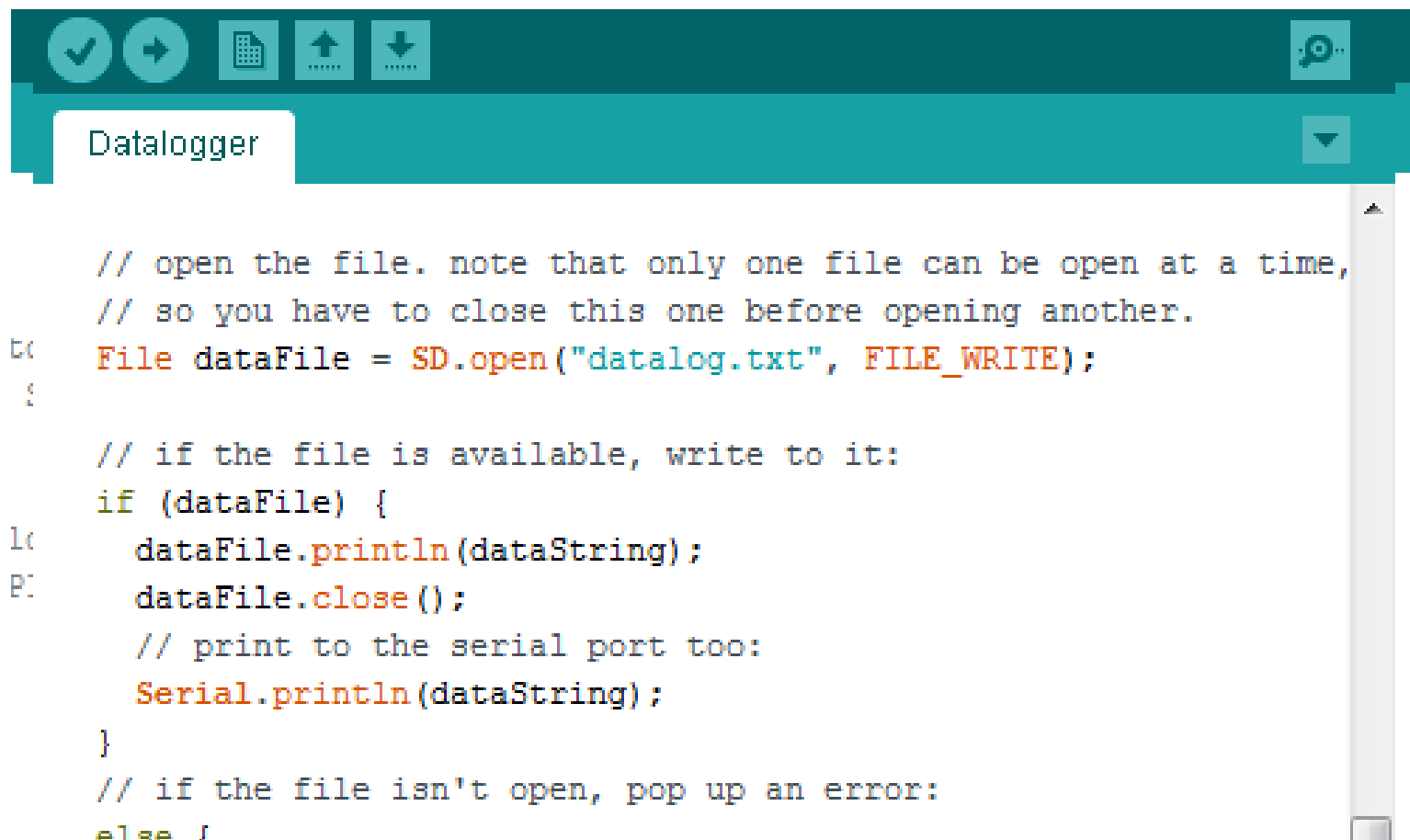
Redundancja danych – karta SD

- Łącze radiowe jest zawodne
- Warto zabezpieczyć się zapisując dane w samym CanSacie, tak aby odzyskać je po lądowaniu
- W pamięci CanSata można też zapisać większą ilość danych (np. częściej próbkować czujniki)
- Płytką główną CanSat Kit zawiera w sobie złącze na karty micro SD podłączone do mikrokontrolera interfejsem SPI



Redundancja danych – karta SD

- dostępna biblioteka do użycia kart SD z Arduino
- przykład **Datalogger**

A screenshot of the Arduino IDE interface. At the top, there is a toolbar with icons for checking, running, uploading, and downloading. Below the toolbar, a tab labeled "Datalogger" is selected. The main area displays C++ code for an SD card logger. The code includes comments about opening and closing files, and uses the SD and Serial libraries. The code is as follows:

```
// open the file. note that only one file can be open at a time,  
// so you have to close this one before opening another.  
File dataFile = SD.open("datalog.txt", FILE_WRITE);  
  
// if the file is available, write to it:  
if (dataFile) {  
    dataFile.println(dataString);  
    dataFile.close();  
    // print to the serial port too:  
    Serial.println(dataString);  
}  
// if the file isn't open, pop up an error:  
else {
```

Redundancja danych – karta SD

- Szczegółowe informacje w rozdziale „**5.6 Używanie karty micro SD**” dokumentacji zestawu CanSat Kit
- Dodatkowa dokumentacja:
 - <https://www.arduino.cc/en/Reference/SD>
 - <https://www.arduino.cc/en/Tutorial/ReadWrite>
 - <https://www.arduino.cc/en/Reference/SDCardNotes>

Praktyczne wskazówki

- używaj kart o stosunkowo niewielkiej pojemności < 32 GB
- karta powinna być sformatowana na system plików FAT16 (ew. FAT32)
- zapisywane pliki muszą mieć nazwę w formacie 8.3 czyli maksymalnie 8 znaków nazwy, 3 znaki rozszerzenia np.: **cansat.txt**
- używaj metody **flush()**, aby uniknąć utraty danych