

Using Files on the Lego EV3 with RobotC

A set of functions has been written so that you can use ASCII files with the Lego EV3 robot. To upload and download the files, you should use the file utility found under

Robot > Lego Brick > File Management Utility

Data files on the EV3 are located in the rc-data directory.

The code to use each of the functions (and the corresponding C++ code) is found after the list of functions, below. The functions available are:

```
bool openWritePC(TFileHandle & fout, char* name);
bool openReadPC(TFileHandle & fin, char* name);
bool closeFilePC(TFileHandle & fileHandle);
bool writeCharPC(TFileHandle & fout, byte charmsg);
bool writeEndlPC(TFileHandle & fout);
bool writeTextPC(TFileHandle & fout, string const & textmsg);
bool writeLongPC(TFileHandle & fout, long number);
bool writeFloatPC(TFileHandle & fout, string const & numFormat, float number);
bool writeFloatPC(TFileHandle & fout, float number);
bool readCharPC(TFileHandle & fin, char & charmsg);
bool readTextPC(TFileHandle & fin, string & result); // a maximum of 20 characters
can be read
bool readIntPC(TFileHandle & fin, int & number);
bool readFloatPC(TFileHandle & fin, float & number);
```

Before Coding

C++:

N/A

RobotC:

Download PC_FileIO.c

Include Files

C++:

```
#include <fstream>
using namespace std;
```

RobotC:

```
#include "PC_FileIO.c"
```

Open Input File

C++:

```
ifstream fin("readFile.txt");
```

RobotC:

```
TFileHandle fin;  
bool fileOkay = openReadPC(fin, "fileRead.txt");
```

Open Output File

C++:

```
ofstream fout("writeFile.txt");
```

RobotC:

```
TFileHandle fout;  
bool fileOkay = openWritePC(fout, "fileWrite.txt");
```

Close File

C++:

```
fileName.close();
```

RobotC:

```
closeFilePC(fileName);
```

Read From File

C++:

```
char c;  
string s;  
int i;  
double d;  
fin >> c >> s >> i >> d;
```

RobotC:

```
char c;  
string s;  
int i;  
float f;  
readCharPC(fin, c);  
readTextPC(fin, s);  
readIntPC(fin, i);  
readFloatPC(fin, f);
```

Write to File

C++:

```
char c = 'A';  
string s = "Hello";  
int i = 3;  
double d = 5.6789;  
fout << c << endl << s << endl  
      << i << endl << d << endl;  
fout << fixed << setprecision(2) << d;
```

RobotC:

```
char c = 'A';  
string s = "Hello";  
int i = 3;  
float f = 5.6789;  
writeCharPC(fout, c);  
writeEndlPC(fout);  
writeTextPC(fout, s);  
writeEndlPC(fout);  
writeLongPC(fout, i);  
writeEndlPC(fout);  
writeFloatPC(fout, f);  
writeEndlPC(fout);  
writeFloatPC(fout, "%.2f", f);
```