

```
#include "Aria.h"

int main(int argc, char** argv) {
    // mandatory init
    Aria::init();

    // set up our parser
    ArArgumentParser parser(&argc, argv);
    // set up our simple connector
    ArSimpleConnector simpleConnector(&parser);
    // robot
    ArRobot robot;
    // load the default arguments
    parser.loadDefaultArguments();

    // parse the command line... fail and print the help if the parsing fails
    // or if the help was requested
    if (!simpleConnector.parseArgs() || !parser.checkHelpAndWarnUnparsed()) {
        simpleConnector.logOptions();
        exit(1);
    }
    //end of clipped data;

    // set up the robot for connecting
    if (!simpleConnector.connectRobot(&robot)) {
        printf("Could not connect to robot... exiting\n");
        Aria::exit(1);
    }

    // start the robot running, true so that if we lose connection the run stops
    robot.runAsync(true);

    robot.comInt(ArCommands::ENABLE, 1);

    //end of required includes, begin custom code
    robot.setHeading(0);
    while (1) { if (robot.isHeadingDone()){break;} }
    printf("at 0 deg\n");

    robot.move(700);
    while (1) { if (robot.isMoveDone()){break;} }

    robot.setHeading(90);
    while (1) { if (robot.isHeadingDone()){break;} }
    printf("at 90 deg\n");

    robot.move(700);
    while (1) { if (robot.isMoveDone()){break;}}
```

```
robot.setHeading(180);
while (1) { if (robot.isHeadingDone()){break;} }
printf("at 180 deg\n");

robot.move(700);
while (1) { if (robot.isMoveDone()){break;} }

robot.setHeading(270);
while (1) { if (robot.isHeadingDone()){break;} }
printf("at 270 deg\n");

robot.move(700);
while (1) {if (robot.isMoveDone()){break;} }
//end of square

robot.setHeading(0);
while (1) { if (robot.isHeadingDone()){break;} }
printf("at 0 deg\n");

}
```