

# Getting Started with the RCX and IC

KISS Institute for Practical Robotics  
1818 W Lindsey, Bld D Suite 100  
Norman, OK 73069 USA  
1-405-579-4609

# Using this documentation

- This is a getting started manual.
- More complete software documentation is available under the help menu when running IC
- The software examples in this manual are for illustration only. They may not be absolutely correct or complete.

# Setting Up

- **IC** runs partly on the PC and partly on the robot board
- The **IC** editor can be used without an attached robot board
- Code can be checked for syntax errors
- To check for logic errors you:
  - Can simulate execution of your program using built in simulator, or
  - Attach a robot and run your program
- The RCX needs to have firmware loaded before programs can be downloaded onto robot

# To Install IC

- On a Mac OSX (10.1 and higher)
  - Double click on InteractiveC5xxx.tar.bz2 file
    - The IC5 folder can be placed in your Applications folder, or anywhere else convenient
    - Note: keep the app and the library folders in the same IC5 folder (programs you write can be kept wherever you wish)
- On Windows (Win 98 and higher)
  - Double click on InteractiveC5xxx.exe
    - IC5 will be added to your program menu
    - An IC5 shortcut will be placed on your desktop
- On Linux
  - Contact support <at> kipr.org

# Lego RCX

- Simple controller
  - Three sensor inputs
  - Three motor ports
  - Communicates to your computer via the IR tower
  - RCX powered by 6 alkaline AA batteries
  - IR tower powered by 9v alkaline battery
  - Gray DB9 to DB9 cable connects tower to serial port

# RCX

View button:

`view_button()`  
or `start_button()`  
(displays sensors port  
values if program is NOT  
running)

Prgm button:

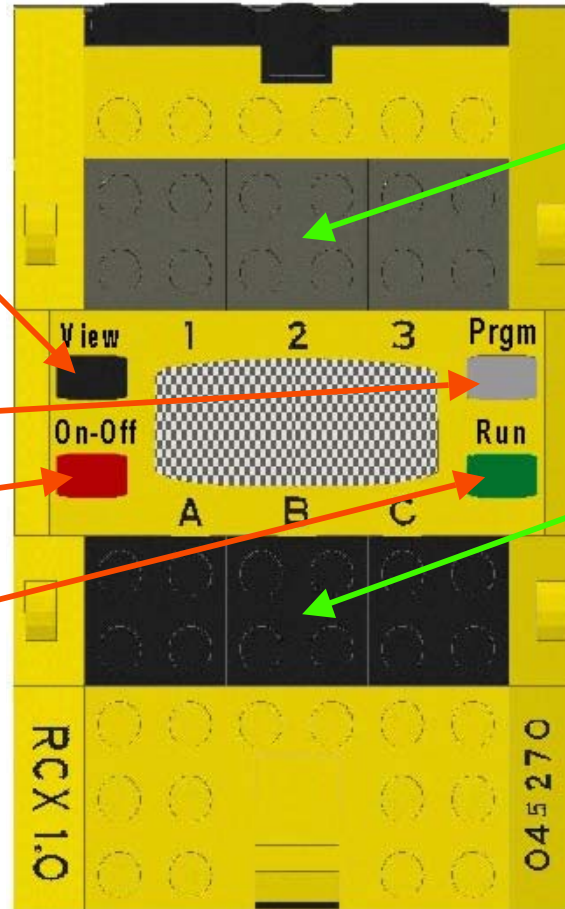
`prgm_button()` or  
`stop_button()`

Power button

turns RCX On and Off

Run Button

runs the main function



Ports 1, 2, & 3 are the  
sensor ports

Ports A, B & C are the  
motor ports (referred  
to in code as A, B &  
C (1, 2, & 3 also  
work) e.g., `fd(A)` ;  
or `fd(1)` ; turns on  
the motor on port A)

# RCX Checklist

- Lego RCX
  - Main Board
  - Battery Box
  - 6 AA batteries
- RCX to IR tower cable
- IR tower
  - 9V battery

# IR Tower Battery

Exposed connector means Bad



- Be sure to insert the battery all the way (**this is the cause of most IR tower failures!**)
- Connectors fit into the left wall
- **If your communication does not work, check battery**

Good



# RCX Setup

To your serial port

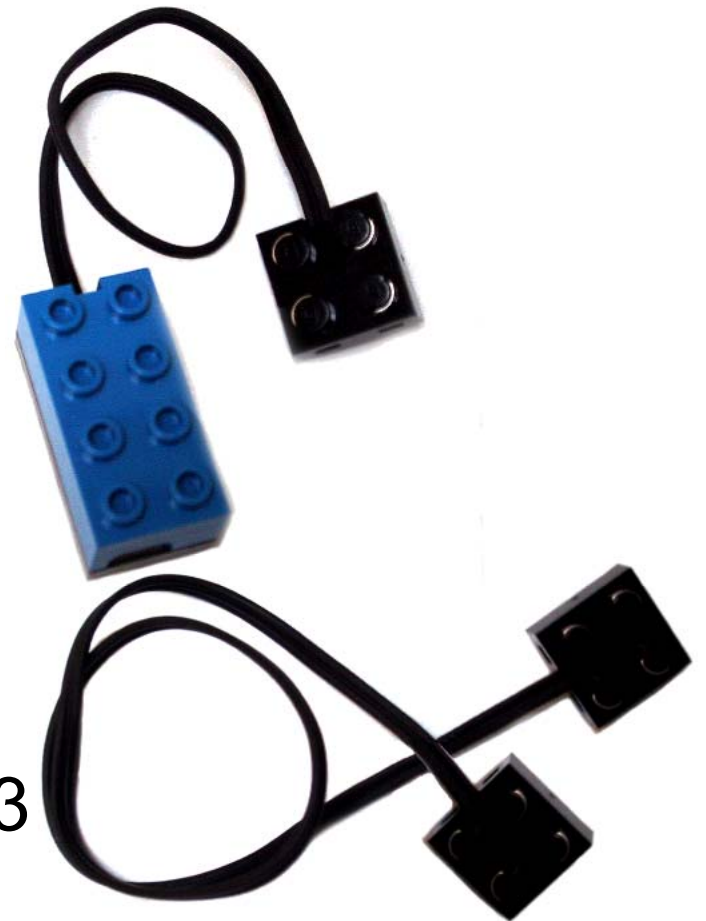


# Download the Firmware

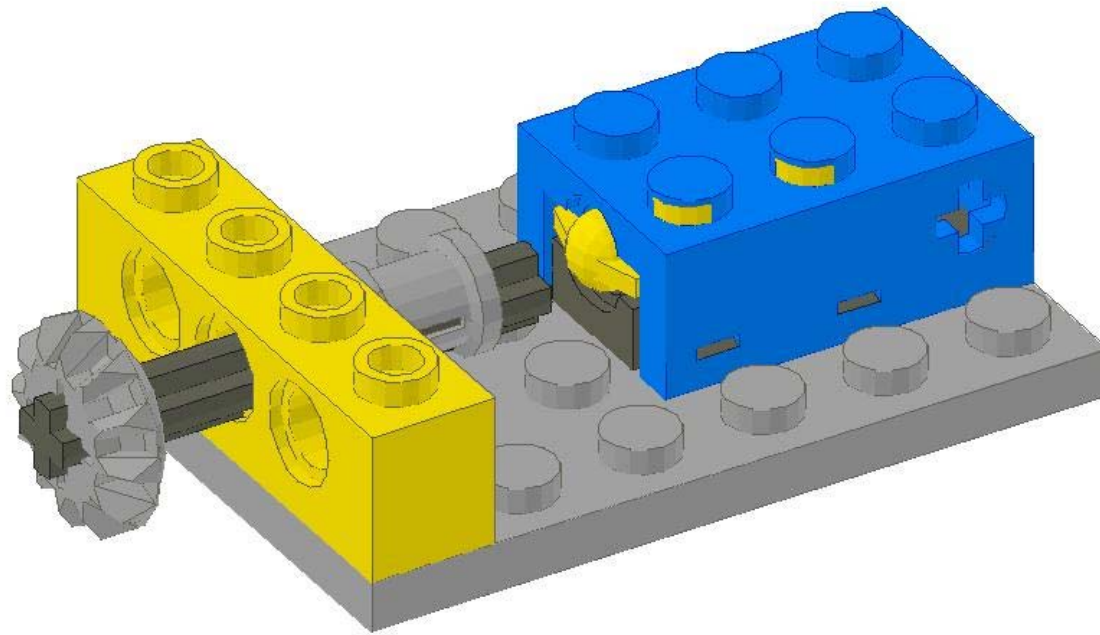
- Make sure your RCX is on and facing the IR tower which is connected to your personal computer via serial port cable (and USB to serial adapter if needed). Note: RCX USB IR tower does not work with IC
- Start IC
- Select the RCX controller
- Select the appropriate serial port
- Click on **Download Firmware** button if needed
- It is a good idea to put a box over the RCX and IR tower while downloading to shield the transmission from stray infrared signals that may be present in the room
- It may even be advisable to put a piece of black tape over the IR end of the RCX after the download is done!
- Note that the firmware download takes several minutes

# RCX Sensors

- Touch sensor
  - (connector wire must cover top front four knobs)
- Light and reflectance sensor
- Connect to RCX with Lego to Lego wire
- Sensors plug into ports 1,2 or 3



# A Simple Bumper



# Examples of Motors that Work with the RCX

- Black gear motors have built-in connector and are medium speed and high torque
- Red & Gray motors use detachable wire
- Gray motors are high speed and medium torque
- Red motor is light, low speed and low torque
- Motors plug into ports A,B or C



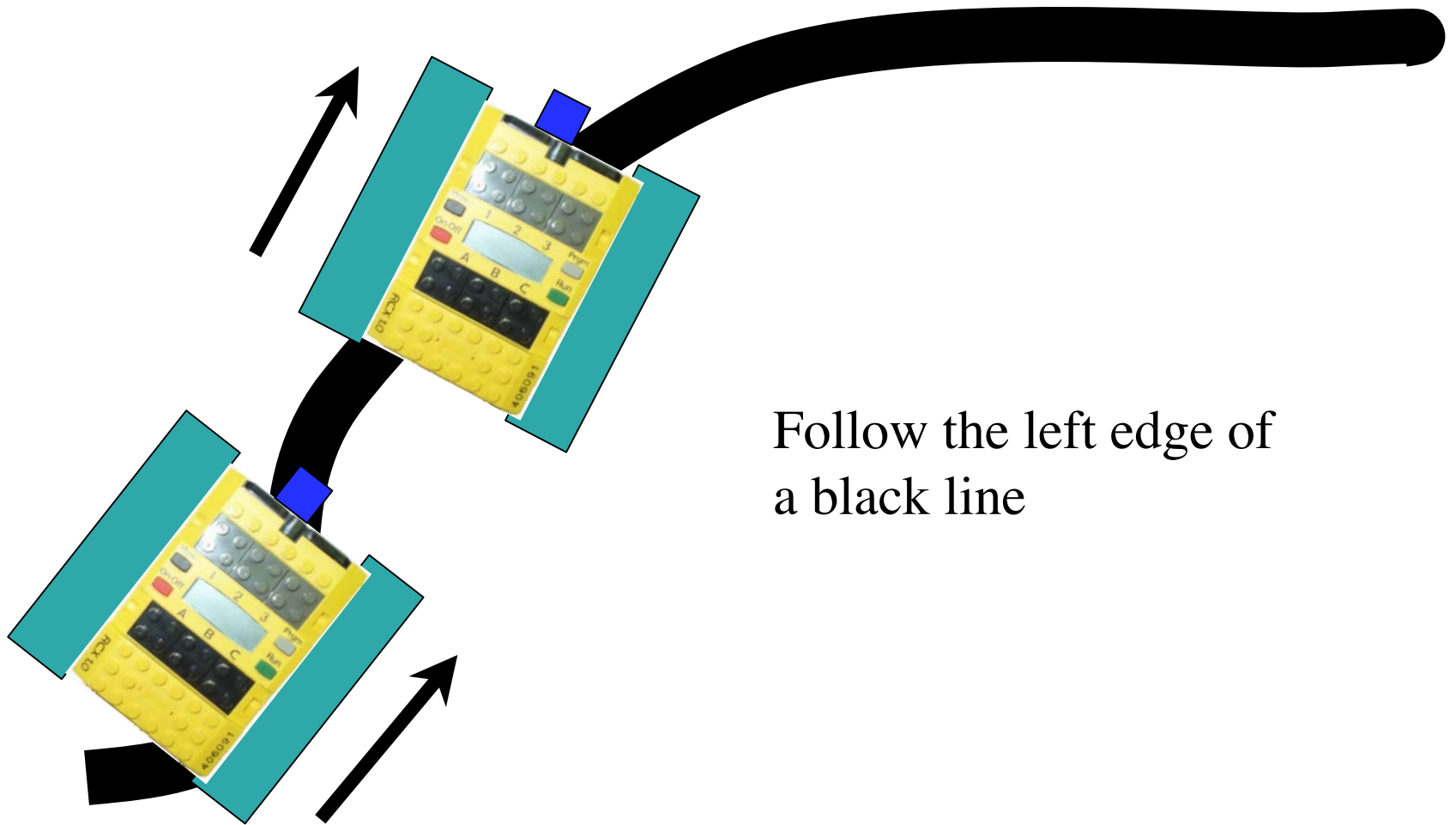
# Testing your RCX

- Using Lego to Lego wires, attach your Lego touch sensors to ports 1 and 2
- Attach your light sensor to port 3
- Using Lego to Lego wires, attach motors to ports A, B and C
- Click on **IC's OPEN** button
- Load the file RCX/rcxtest.ic located in the **IC** program folder
- Turn on the RCX and orient it towards the IR tower
- Click on **IC's DOWNLOAD** button -- the program will now be downloaded to the RCX
- After download is complete press the RUN button on the RCX
- Follow the scrolling instructions on the RCX and press the touch sensors and block, unblock the light sensor -- these actions cause changes in the motor speeds

# RCX Cool Functions in IC

- **battery\_volts** ( ) returns the level of the RCX batteries
- **brake** (*n*) stops motor *n* (e.g., 1, 2 or 3) quickly (more quickly than **off** (*n*) )
- **allbrake** ( ) same as above
- **light** (*p*) returns value of light sensor, with emitter on, connected to port *p*
- **light\_passive** (*p*) returns value of light sensor connected to port *p*
- **poweroff** ( ) turns off the RCX

# Line Following



Follow the left edge of  
a black line

# Line Following (in English)

- Human: Place robot so sensor is over black line, with robot aimed along the line; press run. Press **prgm\_button** to stop
- Program: measure the light sensor on black
  - store this in the variable named **dark**
  - while the **prgm\_button** is not pressed
    - if light sensor reads less than **dark-1** (bright), turn right (left tread forward, right stopped)
    - otherwise turn left

# Example Program

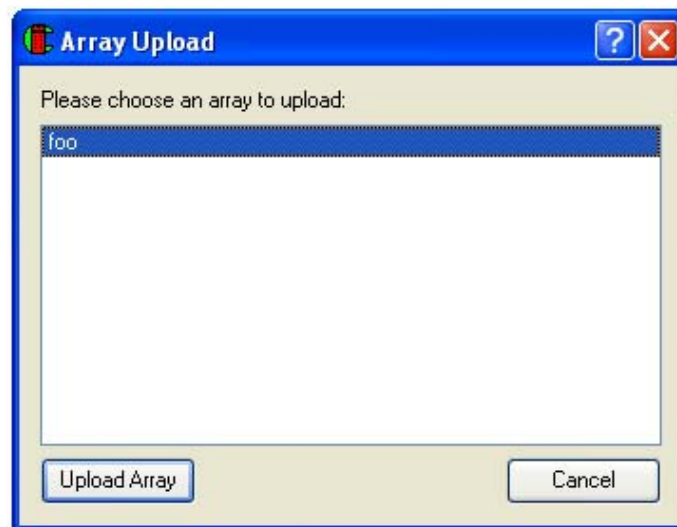
```
/* This main function demonstrates some nice features of having the
robot wait until it is in the right place and the user is ready,
before actually starting to move. The functions turn_right and
turn_left are left as an exercise */
void main() {
    int black_line;      // place to store best sensor value of blk line
    black_line = light(3);
    while(view_button()==0) { // tell user what to do, repeatedly
        black_line = light(3); // read blk and display it; wait for Strt
        printf("blk V");
        sleep(.05);      // wait briefly to stabilize display
    }
    printf("P end"); // so user knows what
    while(prgm_button()==0) { // to do to stop
        if(light(3) < black_line - 15) {
            turn_left(); // call the function to turn the robot
        } // back towards the black line
        else {
            turn_right(); // turn the robot away from the line
        }
    }
    ao(); // make sure all of the motors are off
    printf("All done\n"); // let the user know you are done
}
```

# Global Arrays

- Global arrays are still in memory after the program exits
  - **IC** doesn't clear the stack until next execution of program entry point
- Global arrays can be uploaded from the RCX and placed in spread sheet
- You can also list global variables, functions, etc...

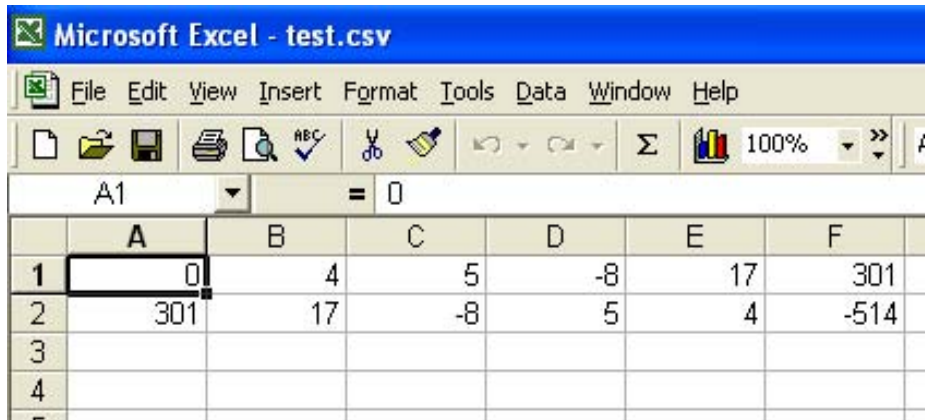
# Uploading Global Arrays

1. Run a program to create or gather data in an array
2. After the program is finished or paused, connect RCX to computer
3. From the Tools tab choose Upload Array
4. Pick the array to upload



# Uploading Global Arrays (cont.)

5. Click Upload Array
6. Choose how to save the data
7. View data

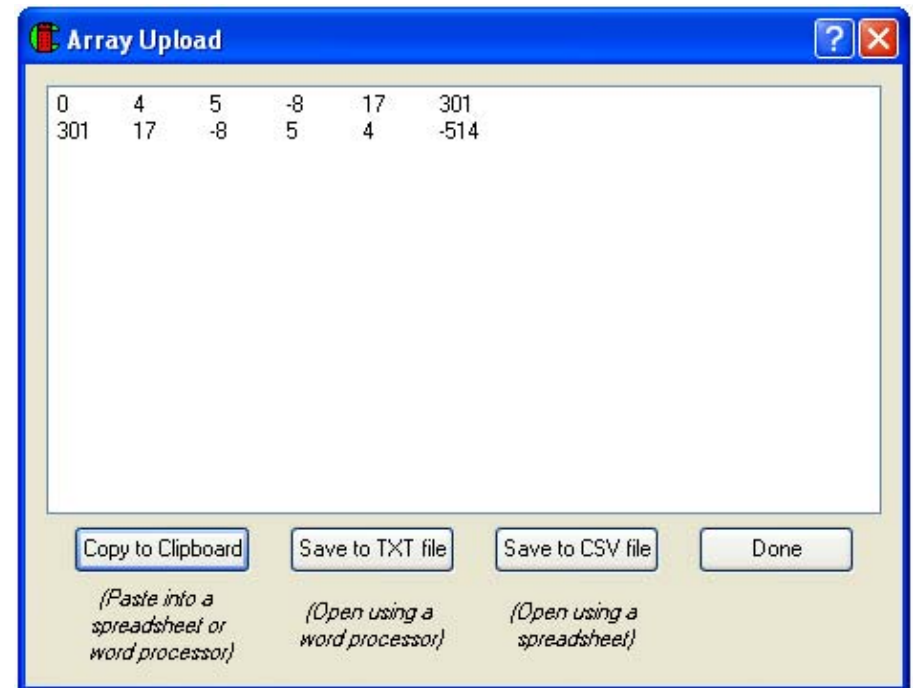


Microsoft Excel - test.csv

File Edit View Insert Format Tools Data Window Help

A1 = 0

	A	B	C	D	E	F
1	0	4	5	-8	17	301
2	301	17	-8	5	4	-514
3						
4						



# Notes on Uploading Global Arrays

- After uploading program to RCX, leave IC window open
- The Array must be a Global Array
- Only one array can be uploaded at a time, but you can go back to upload multiple arrays
- To open a CSV file in Excel, you must select “Text Files” file types in the Open file window