# Colony - Enhancement #323

## Wireless bootloading

01/28/2009 06:33 PM - Kevin Woo

| | | | |
|---|---|---|---|
| **Status:** | Fixed | **Start date:** | |
| **Priority:** | Tabled | **Due date:** | |
| **Assignee:** | Ryan Cahoon | **% Done:** | 90% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | Misc. | | |

**Description**

Reference design
http://www.sparkfun.com/commerce/tutorial_info.php?tutorials_id=122

### History

**#1 - 03/12/2009 01:05 PM - Kevin Woo**

*- Status changed from New to Feedback*

*- Assignee deleted (Unassigned Ticket -)*

**#2 - 03/21/2009 12:54 PM - Rich Hong**

*- Category set to 29*

**#3 - 04/20/2009 10:47 AM - Kevin Woo**

The sparkfun method uses a custommized windows program to do it. They could not get it to work with avrdude. Also, they use the RTS/DIO pin to reset the avr in order to put it into bootloader mode. According to initial research done by Chris, our SBees have a firmware bg that disables RTS/CTS.

**#4 - 09/26/2009 04:54 AM - Ryan Cahoon**

*- Status changed from Feedback to Fixed*

*- Assignee set to Ryan Cahoon*

*- % Done changed from 0 to 90*

Prototype code added in SVN revision 1428. New bootloader code supports both wired and wireless programming. Requires use of new loader client instead of avrdude, but code can be compiled cross-platform. No hardware changes are required, but AVRs requires an avrisp or similar to flash new bootloaders.

Code download works on all programs tested thus far, but "guinea-pig" testers should probably use the new loader for a while before it could become the default loader.

Further ideas:
--Create a hook on a special command in libwireless which sets the robot into program mode, so no physical contact with the robots is required to program them
--Allow robots to program each other, so code updates could be pushed to a single robot, which could then recursively push it to others