

With minor setbacks, we were able to get the Freeduino/Duemilanove wirelessly programmed with Synapse Wireless RF100 RF engines at the speed of 57600 baud. But when moving over to trying the UNO, it seemed to be much more of a beast to program.

One big issue we ran into was the timing of the reset signal needed by the Optiboot bootloader on the UNO ATMEGA328p chip. The reset timing seemed to be quite precise and we had a hard time figuring it out. We eventually did figure it out by capturing the signals with our Saleae

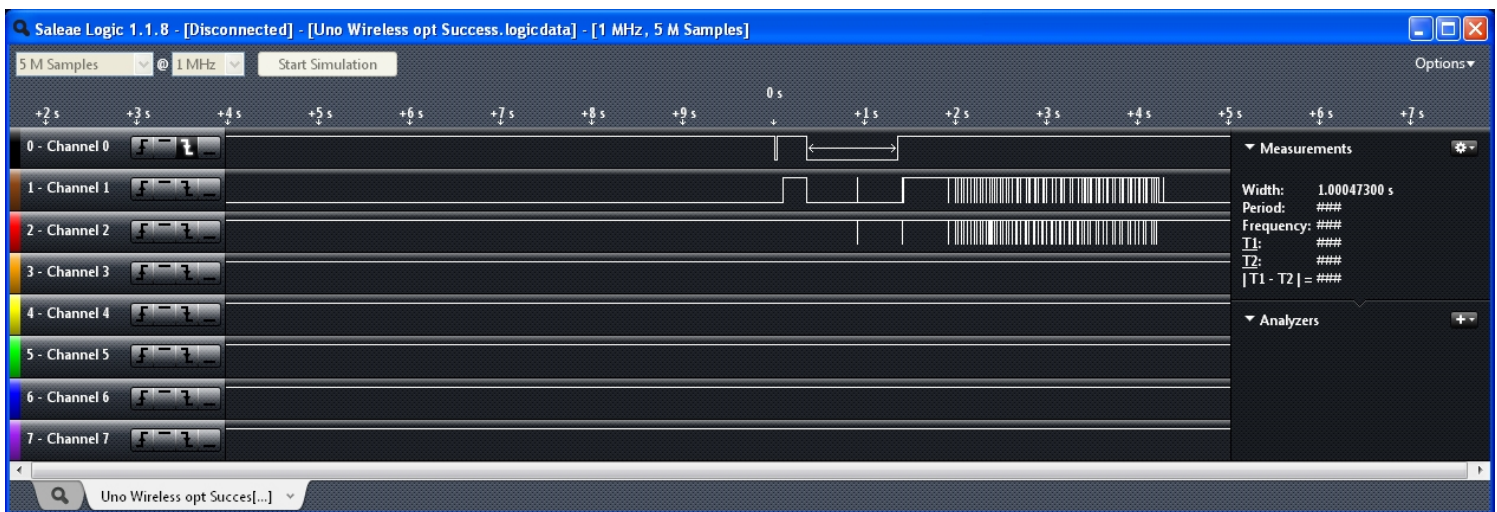
logic analyzer and manually toggling the reset switch, it was quite the trial and error experience.

Here is what the normal UNO Reset and Serial transmission look like when programming with a USB cable:

Channel 0 (Black) = Reset

Channel 1 (Brown) = Arduino UNO TX pin/Synapse node RX pin

Channel 2 (Red) = Arduino UNO RX pin/Synapse node TX pin



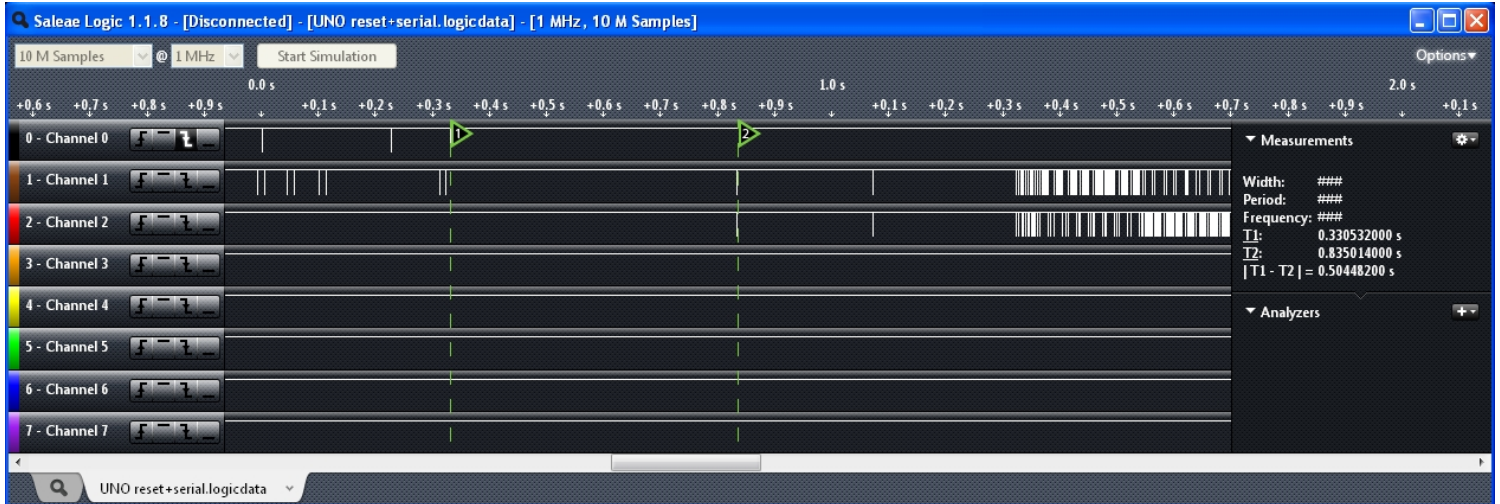
Sparkfun, and others, actually slightly modified the original Arduino optiboot bootloader to combat the precise reset timing needed. They modified the watch timer timeout from 500ms to 1S to allow for a less precise reset signal to be received before serial packets were transmitted/received by the UNO chip. They did this in order to receive an auto-reset, from the DTR line of an FTDI breakout, and thus be able to program the UNO automatically, otherwise a manual reset (a physical push of the reset button) would be in order. If you'd like to read about Sparkfun's Optiboot changes the info is [here](#).

We found that although the specs for the RF100s did indeed state that they can do transfer rates up to 250kbps data rate, we did

not know if this was practically achievable over the air. Turns out 115200 baud is achievable when you set the baud to 1 (i.e. `initUart(1, 1)`).

Using Arduino1 software with the new "arduino" serial programming protocol instead of the old "stk500" protocol (in earlier Arduino versions) we can achieve a correct reset with the code we have written for the Synapse RF engines. We were able to code the synapse nodes to extend the reset pulse to 1S and properly time it to have it finish resetting during the Optiboot's second attempt at communication, we did try to line it up with the first attempt but could seem to get it to work.

Here's what our signals looks like when wirelessly programming the [Arduino UNO](#) with the synapse nodes:



The measurement of 1S represents the 1S reset delay that we programmed into the Synapse node.

Please note: Arduino-0022 and Arduino1 have a difference in the boards.txt file being that they have a different serial programming protocol from one to the other. Instead of using the 'STK500' protocol (like in Arduino-0022) it is now changed to 'Arduino' protocol.

With larger file sizes and greater transmission distances an established link is harder to maintain as the data throughput is at 115200 baud. That being said, we are able to upload code to an UNO from indoors in our lab to a park bench across the street (~250feet away). At Solarbotics Headquarters we have been getting very good results but in different environments mileage may vary. Please be aware that our code isn't perfectly refined as the Non-Volatile Parameters (Nvparams) probably aren't exactly calibrated, and we don't have enough experience to know which values

would be best we just selected values that gave us the best results. The following parameters can be adjusted to maximize serial transmission:

Buffering Timeout (13) Default value:5

Buffering Threshold (14) Default value:75

Inter-character Timeout settings (15) Default value:0

One major advantage to using the Synapse Wireless RF engines as opposed to competing RF modules that can program Arduino UNOs is that they have a large transmission range, they're rated for more than 20 times the distance of other competing modules.

Note: All these results were done with the RF100 node with the F type PCB antenna (RF100PC6), if the SMA antenna version (RF100PD6) was used we could have potentially achieved even better results.

*Screenshots created with Logic 1.1.8 and the [Saleae Logic Analyzer](#), sold at [Solarbotics.com](#)