

Bliptronome!

Bliptronic 5000 to Monome Clone Conversion
Getting Started with the software and Testing

Wil Lindsay

www.straytechnologies.com
wil@straytechnologies.com

The Bliptronic 5000 is a low cost matrix synthesizer created and sold by ThinkGeek™ . The design affords very simple adaptation for other purposes including as a real-time multimedia controller. The “Bliptronome” design replaces the original device's synthesis module with a minimal version of the Arduino microcontroller platform. As such, the Bliptronome firmware allows the newly created device to serve as a button/led matrix platform able to work with applications designed for the Monome platform, and independently written OSC, Midi, and Serial based applications.

It is especially relevant to music and live media performers as it allows individuals to create their own production tools in software such as Processing, Max/MSP, Puredata, Python, and other development platforms, while maintaining a tangible hardware interface.

Getting Started with the Software:

For those new to the Monome paradigm, it can be thought of as a completely programmable controller, where every button, LED and knob can be individually controlled as the user or application developer sees fit. It is compatible with a wide range of software platforms and operating systems and could be used as an extension to many applications (not just music!)

In order for a Monome, Arduinome or Bliptronome to “do anything” it needs a few basic components explained in detail below:

1: usb connection to a computer

The “FTDI” usb-to-serial cable provides two functions: It facilitates communication with the host application, and also provides a 5-volt power source for your device. In the case of the Bliptronome, that means no batteries!

The business-end of the FTDI cable attaches to the Bliptronome via a 6 pin connection recessed on the side of the unit. It is important that the cable not be attached “upside-down” (no known damage occurs if this is accidentally reversed, though the unit will not function this way). In the case of the black FTDI cable the black wire must be in the right-most pin when looking down at your unit. In the case of the red “BUB board” plug, the surface mounted components must be on the top side of the little board when attached to the Bliptronome.

A correctly attached plug begins the Bliptronome's fast “LED scan sequence,” alerting the user that the Bliptronome has properly powered up.

2: an FTDI device driver

A software device driver found at <http://www.ftdichip.com/Drivers/VCP.htm> is needed for your computer's operating system to correctly identify the Bliptronome as a usb device. Drivers are referred to as a “VCP” or “Virtual Com Port” driver, and are provided for most useful operating systems on that web page. The cable is seen as an “FT232R” Device and has a device name of “a40h-5xx” where “xx” is any number. Use your operating system's typical device installation procedure or “wizard” to install this driver. A full reboot after driver installation will make life easier on both OSX and Windows machines.

3: a software “bridge” driver

The Monome family of devices are most useful with a “bridge” driver to convert the device's native communication protocol to something more useful to other applications such as Midi or OSC. Those of a “hacker” background can likely program the device via serial, and have already thrown away these instructions before arriving at this sentence. Linux users have an option referred to as “serial-pyio” available here: <http://sourceforge.net/projects/serial-pyio/>

For the rest of us, a program referred to as “ArduinomeSerial” is the current best choice and available halfway down this page for OSX and WIN:

<http://sourceforge.net/projects/arduinome/files/Arduinome/>

Once installed and running, ArduinomeSerial sees your plugged in Bliptonome by its device name “a40h-5xx.” Other options in this software are as follows for testing sake.

I/O Protocol : usually “OpenSoundControl” though other options are available for the experienced
Host Address: “127.0.0.1” (this is what your computer calls itself on a network).

Host Port: usually “8000”

Listen Port: usually “8080” – these 2 numbers can change, but must match the application you are using.

Device: “a40h-5xx” as mentioned above. The software can see 2 or more at a time with different device names.

Cable Orientation: this can be changed by the user. On the Bliptonome “top” or “up” refers to the device's normal position with knobs to the right. NOT the plug port location.

Address Prefix: This is determined by the host application and needs to match that application for proper OSC communication. (“/test” is common for the test patch mentioned below)

The next 4 options are usually used for multiple units and should remain at “0” for a single Bliptonome.

ADC Encoder states : On the Bliptonome the 4 knobs are referred to (top to bottom) as ADC 0, ADC 1, ADC 2, and ADC 3. These 4 positions must have check-marks in them, with none in the ENC boxes. Unplugging the device with ArduinomeSerial may require the ADC boxes to be “re-checked.” along with other problems.

4: a host / application

So close! Now the good part! There are a myriad of applications that will work with the Bliptonome. Anything “Arduinome compatible” will work. PLEASE email me if you find an exception. So far the only “Monome” apps I have seen not working are those designed for a protocol called “SerialOsc.” Most (but not all) applications have been written in high-level user programmable software such as Max/MSP, PureData, or Chuck. As such you will need the host IDE to run each respective app. Covering all of these apps is FAR beyond the scope of this document. But, here is the wiki list to get you started:

<http://docs.monome.org/doku.php?id=app> (note that the top half of this list are SerialOsc patches, and NOT for the Arduinome clones like our Bliptonome. All of the apps from halfway down the page: http://docs.monome.org/doku.php?id=app#apps_that_have_not_been_confirmed_as_working_without_monomeserial

http://docs.monome.org/doku.php?id=app#apps_that_have_not_been_confirmed_as_working_without_monomeserial

Seem to work at this time. Other apps will almost always have a non SerialOsc version that works fine for us.)

For testing purpose we will use Monome “test” with the Max/MSP runtime.

You will need the Max 5.1 Runtime from here: <http://cycling74.com/downloads/> if you are not already a Max/MSP guru. Download. Install.

Our first patch will be this:

<http://docs.monome.org/doku.php?id=app:monomebase>

(again, make sure you download monome_base_4.4.zip NOT the SerialOsc version)

Once everything is downloaded and installed:

1. plug in the usb/FTDI cable & watch the left-to-right scanning power-up sequence on the Bliptronomie.
2. Open ArduinomeSerial and verify the settings selected above.
3. Start Max Runtime
4. From Max Runtime File/Open a “patch” called monome_test.mxb or monome_test
5. Click the button that says “<-- you must click here...” This automatically changes the address prefix in ArduinomeSerial to “/test” and allows proper communication.
6. Change “Pairing” option to “PRESS” and start mashing buttons and twiddling knobs.

This patch demonstrates all of the functions: Experiment! Buttons light the software button grid. Clicking in the software lighting grid lights LEDs. Knobs change ADC levels.

Check out more Apps! Write your own! Don't forget to eat & sleep!

Final Notes on the Bliptronomie

The Bliptronic 5000 is a \$50 device. I've found it to be very hardy, though I expect overall \$50 build quality. This may be a gateway device to the device's bigger siblings.

SerialOsc is a newer protocol which excludes our Arduinome based Monome clone devices. Despite all of the scramble to update, I've found that ArduinomeSerial is VERY stable and allows cool things like flipping knob orientation and stringing lots of blips together, which serialOsc currently does not. Almost every patch I've found has a “MonomeSerial” version which will work for us, and is not inferior to the newer “SerialOsc” versions of the same patch. (ex: Polygome .98 ROCKS!)

There may be a slight cross-talk in the knobs. This was a design trade-off and can be eliminated only through reduced sensitivity and range in the pots. My design choice was for more sensitivity.

It's true. The “Play” button does nothing. There is one unused pin left on the Arduino and it could be modified as a 65th button, a shutdown switch, or wired to explosives. I leave that to you.

Though this is an open source project; I have done most of the heavy lifting to date. If you are feeling helpful, friendly, wealthy, or appreciative: I'd love a pat on the back. Send me a Blip based song, help me with code updates or suggestions, share, document, buy a synth at my little web shop, come over and walk my Corgi pup for a few minutes (I need a break!), or share a beer with me.

wil@straytechnologies.com

And yeah, I can help and answer questions too!

Software Postramble:

Bliptronome_V2 by Wil Lindsay : February 16th 2010

<http://www.straytechnologies.com> for hardware mods, build information & port blog

Translation of Arduinome firmware for Bliptronics 5000 designed and sold through Think Geek™

<http://www.thinkgeek.com/electronics/musical-instruments/c4e1/>

The microcontroller code for this project is based and ported from the Arduinome project code "ArduinomeFirmware3_2" written by Owen Vallis & Jordan Hochenbaum and revised by Ben Southall. translation of the 40h protocol designed by Brian Crabtree & Joe Lake, and coded by Joe Lake.
<http://flipmu.com/work/arduinome/>

Project platform created on Arduino microcontroller development platform <http://www.arduino.cc>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

--enjoy!