

Using the Mediatek LinkIt One board with linux and Arduino IDE

June 29, 2018

By piecing together a couple of instructables, and a “answer” from stackoverflow. And a couple of things of my own, I was able to get “Blink” to work on the LinkIt One board using Arduino IDE 1.8.5 - most of the thanks goes to user v-i-s-h from instructables Who provided most of what is needed.

Both of these instructables where written by v-i-s-h.

<https://www.instructables.com/id/Using-LinkIt-One-with-Arduino-in-Linux/>

<http://www.instructables.com/id/Programming-LinkIt-One-in-Linux-no-WiNE/>

They mostly work as written - if you take the 1st one, STEP 1 and STEP 2 then start on the 2nd instructable, when that is done move back to the 1st for STEP 4 and 5. It mostly works.

On Arduino IDE version 1.8.5 you will get compile errors, which can be fixed using the information found here:

<https://stackoverflow.com/questions/33136938/mediatek-linkit-one-error-when-compiling-in-arduino-ide-1-6-6-arm-none-eabi-g>

User Alan Wootton has the correct answer - you will still get compiler warning, but it works.

I got it to work by editing platform.txt

and adding **core/** so that it reads:

{build.path}/core/syscalls\_mtk.c.o

instead of

{build.path}/syscalls\_mtk.c.o

platform.txt was located in:

/Users/awootton/Library/Arduino15/packages/LinkIt/hardware/arm/1.1.17

[share](https://stackoverflow.com/a/34301766/895468)[edit](https://stackoverflow.com/posts/34301766/edit)

answered Dec 16 '15 at 0:04



[Alan Wootton](https://stackoverflow.com/users/4959700/alan-wootton)

**11**1

### STEP 1 - Install the LinkIt One core into the Arduino IDE,

v-i-s-h has provided a core url that will work on linux machines.

https://raw.githubusercontent.com/v-i-s-h/LinkIt-One-Linux-Arduino-Support-/master/package\_vish\_linkItOne\_linux\_index.json

You will have to make a few changes, so you’ll need to go into the hidden “.arduino15” directory that should be in your home folder. You are looking for the packages directory under .arduino15

So it will look something like this:

/home/{user}/.arduino15/packages

You’ll want to find the LinkItOneLinuxArduino directory.

### STEP 2 - in the tools/linkit\_tools/1.1.17 directory

“/home/{user}/.arduino15/packages/LinkItOneLinuxArduino/tools/linkit\_tools/1.1.17”

We need to make the scripts executable by issuing the command

chmod a+x \*.sh

\*\*\*DON’T REPLACE THE platform.txt file yet, there is a new version that uses the updated “packtag.py” tool.

--So now we jump over to the 2nd Instructable.

### STEP 3 (STEP 1) Get the updated packtag.py package.

You have to download the packtag.py from here, <https://raw.githubusercontent.com/Seeed-Studio/Lua_for_RePhone/master/tools/packtag.py>

And you’ll copy or move this file to the linkit\_tools/1.1.17 directory from the previous step.

And you’ll need to make the python script executable.

chmod +x packtag.py

### STEP 4 (STEP 2) Download the new platform.txt

You’ll need to right click on the link and then save as - if it doesn’t have the name platform.txt - rename it.

<https://cdn.instructables.com/ORIG/FF5/HOIC/IH0YIXQX/FF5HOICIH0YIXQX.txt>

Or grab it from the instructable - <http://www.instructables.com/id/Programming-LinkIt-One-in-Linux-no-WiNE/>

Replace the platform.txt found in

“/home/{user}/.arduino15/packages/LinkItOneLinuxArduino/hardware/arm/1.1.17” with the file you just downloaded - remember it needs to be called “platform.txt”

**Important - Restart the IDE!**

Now we can jump back to STEP 4 and STEP 5 from the original instructable.

### STEP 5 - Blink The Built in LED.

Make sure to select the Linkit One from the board manager, look for the blink example (standard Arduino example)

\*\*\* NOTE The LinkIt One doesn’t define LED\_BUILTIN (The board core was written long before this became a standard thing to do) \*\*\*

The easiest way to fix this is at the top of the sketch add the line:

#define LED\_BUILTIN 13

Or can also replace all the LED\_BUILTIN with the number 13.

At this point, this is where the instructable kind of gets a little lost and confusing, So here is what I learned.

You need to save the modified version of your sketch (To compile sketches for this board you need to have a saved copy somewhere that you can get to easy).

On my system, the LinkIt One enumerates as two serial ports, /dev/ttyUSB0 and /dev/ttyUSB1 (of course those could change depending on what else I have plugged in.)

And apparently the device could show up as /dev/ttyACM1 and /dev/ttyACM0

What I found is compile the sketch, you need to have the board plugged in, it will fail to upload, but that is ok because we are not going to compile the sketch the way we would with a “normal” Arduino board.

To compile - click on the “Sketch -> Export” tab in the IDE,

In your sketch directory (This is why you need to save it somewhere), you will find a binary called “app.vxp”

Now we need to switch the LinkIt One from UART mode to MS mode - what I’ve found about this is the battery for the LinkIt One needs to also be installed and on….

(this could just be me).

Besure to make the mode change with the power removed.

Once you restart the LinkIt One, a small 10Mb removable drive should show up.

You are going to copy the “app.vxp” file to the MRE folder on the removable drive.

\*\*\* A ONE TIME STEP IS NEEDED \*\*\*

In the root of the LInkIt One drive you’ll see an ‘autostart.txt’ file, and you’ll need to edit it and point it to the app.vxp file, not the test.vxp file it is pointed at.

This should be on the 2nd line of the file, and it should look something like:

App=C:\MRE\app.vxp

You only need to do that once.

Disconnect your LinkIt One, (make sure battery is off as well, no power) - put it back in UART mode, and you should see the LED blink (or your sketch run on it).

That’s about it.