Turing Machine Expander Port Notes

If you are just looking for information on how to connect one of your expanders to your host module check the other document in this repository called "Turing Machine Expander Hookup Guide".

* * *

At this point I have released three Turing Machine Expanders, including my first product the "Bytes" expander, and so I thought it would be helpful to write down some observations and advice for anyone else who is either confused by the Turing Machine Expansion Protocol or wants to develop their own Turing Machine Expanders. (We will be referring to the Turing Machine Expansion Protocol as TMEP for short for the rest of this document.)

The TMEP is really quite an elegant way to create an environment where many modules can share expanders and many expanders can share hosts. The Turing Machine itself is a great module but it is not necessarily something that has never been done before, even in its documentation Tom Whitwell references as inspiration several other modules such as the Buchla 266 source of Uncertainty that used a similar "Linear Feedback Shift Register" algorithm to create slowly changing sequences. Even within Eurorack LFSR based modules had been released before. It is my belief that the Turing Machine is so popular because a) it was open source and easy for people to build and b) the TMEP allowed users to easily add functionality to their Turing Machines according to their unique needs. It is for this reason that I have chosen to add TMEP host functionality to a few of my own modules and support the TMEP environment as a whole.

There is one factor of the TMEP that is both a benefit and a drawback that needs to be mentioned: it normally uses the same 16 pin IDC cable that older Eurorack modules used for power while not being pin-compatible with a Eurorack Power Bus. Eurorack power and modular interconnects already confuse many users and unfortunately it isn't enough to expect all of them to studiously read the manuals that are supplied for their modules. As such, a handful of my customers have connected either their Turing Machine hosts or expanders directly to their power supply and damaged their modules. While it has overall been a pretty small number of users I still take everyone who loses a module (or has me spend time repairing their modules) very seriously

and so I have come up with a few solutions that will not completely eradicate the risk of a broken module but will at least reduce the risk significantly.

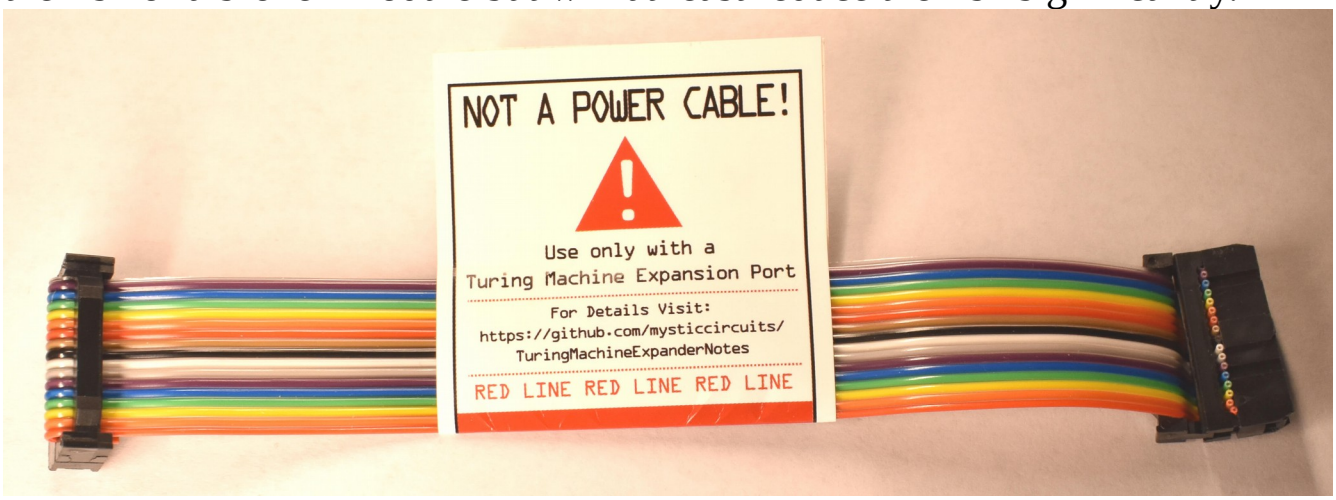

Fig.1 – A cable designed specifically for TMEP modules

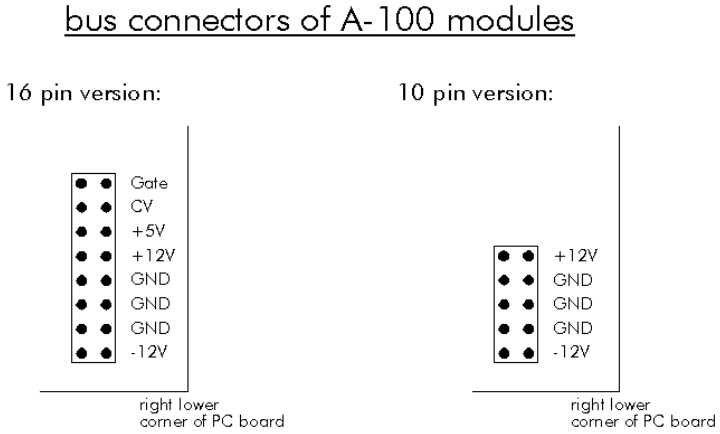

Fig. 2- The 16 and 10 pin Eurorack Power Pin Diagram
(image taken from Doepfer a-100 technical details document)

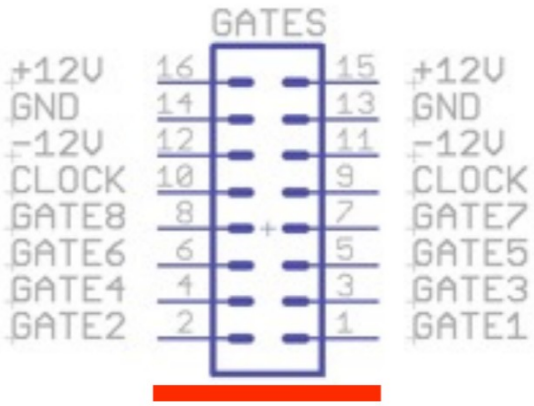

Fig.3 The TMEP Pin Diagram
(image take from Music Thing Random Sequencer Expander Documentation v1)

The first change that I made to indicate that the TMEP port is not intended to be plugged into a power bus board was probably the most obvious - to visually change the cable and to put warnings all over the cable and the PCB. I think that a visual difference in the cable will go a long way to indicate that the user should at least double check that they know what they are doing before plugging the cable in to power even without the sticker. The sticker also refers you to the document you are reading which I can update later to include any information that I discover over the course of releasing products using this protocol so that users can have the most up to date info at their disposal before making a mistake that someone else has made before.

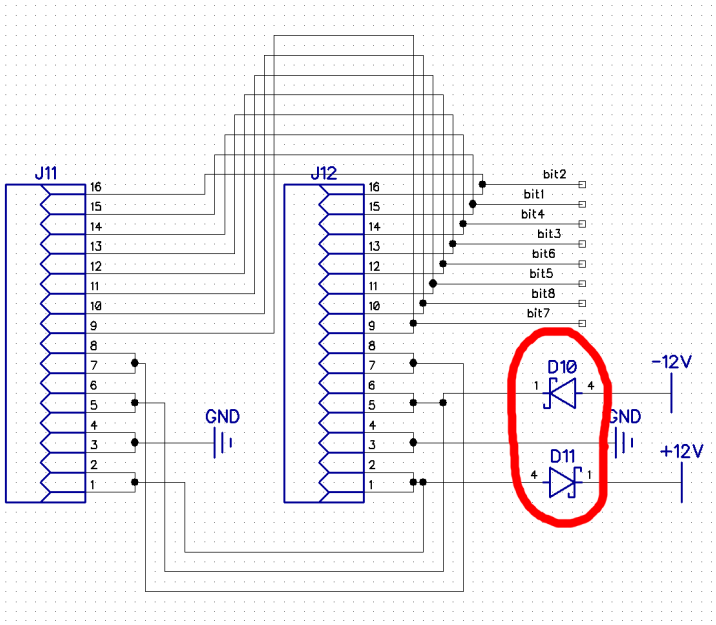

Fig. 4 – Power Rail Protection Diodes

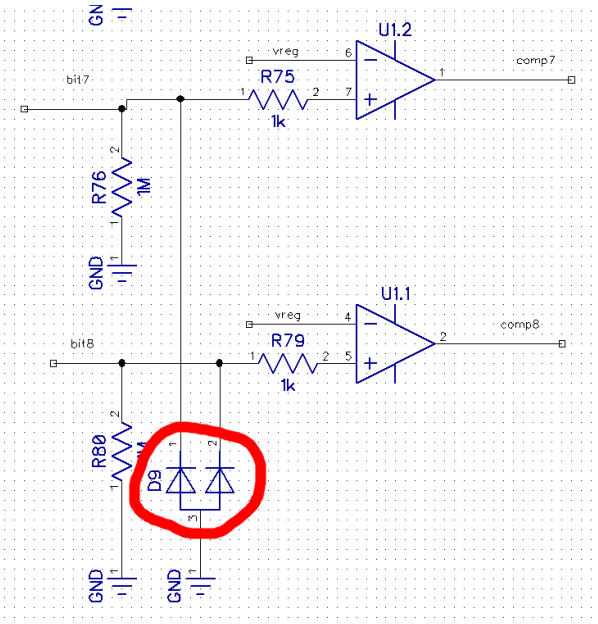

Fig. 5 – Comparator Input Protection Diodes

The next step was to add protection diodes to the expander inputs. This is not a solution I wanted to rely on too heavily because protecting against a one-to-one plugging in a power cable to an expander port kind of situation assumes that the user is plugging it in the right direction on both sides, doesn't have faulty power etc. This is similar to my reasoning as to why I am not a huge fan of keyed power headers: while they prevent many problems in situations where everything is working correctly they can provide false confidence that for example your cables are keyed correctly or that the other side of the cable is plugged in correctly. However in most cases having a shrouded header is better than none and in that way having protection diodes is better than none. In this case the diodes on the left in Fig. 4 will protect the modules power rails when accidentally connecting to the power supply bus in the correct way. Similarly the diodes on the right in Fig. 5 will protect the IC inputs on the bottom row of your expander port from being connected to the negative rail when the expansion port is connected to power.

* * *

Another important problem that folks had with their Turing Machine Expanders was the difficulty in connecting multiple expanders to the same host. Usually I would suggest for folks to buy the floating power distributors from companies like 4MS that made multiple female IDC plugs on a single cable but this was not always an easy solution so with the Bytes Expander I put in duplicate Turing Machine Expansion ports to help mitigate this problem. There were still sometimes issues with cables interfering with each other when both plugged into ports that were adjacent because IDC cables

generally want to flow in a direction determined by their “strain relief” - the plastic housing connected to the port on the cable. For this reason the TMEP cables were designed to have their cable flow in the same direction no matter which way the cable is facing, it allows for easy nesting of one cable on top of another and generally requires fewer kinks in the cable to connect adjacent modules. I understand that this might not be easy to follow in text, for an example of what I am talking about take a look at figure 6 and imagine either that the cables were flowing into each other or away from each other. Both options lead to a mess of cables, so making the cables rest one on top of the other was what I determined to be the easiest solution.

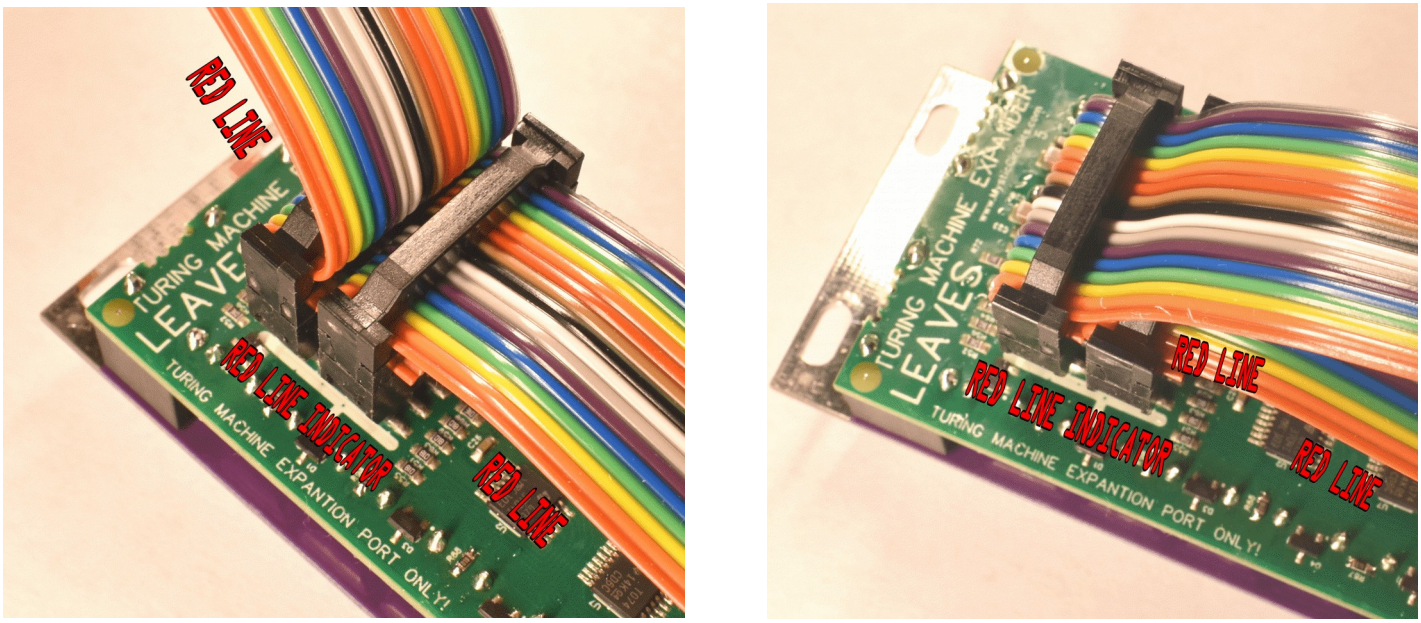

Fig. 6 A and B, cables nesting into eachother, cables resting on top of each other.

The final improvement on the TMEP that I came up with had less to do with protecting the expander or host from damage and more to do with the fact that sometimes connecting a lot of modules together through ribbon cables can cause the modules to behave slightly differently. This is because ribbon cables connected through pin headers multiple times in series will have a higher resistance than either the same modules connected in a star configuration or connected directly through a copper trace on a circuit board and therefor will lose a bit of voltage off of the positive and negative supply rails for each expander connected. This isn’t a huge problem for something digital or using CMOS to process gates because voltage levels do not need to be precise but for something like a note sequencer that has a specified range

of travel over the course of its slider (like Leaves) it presents a problem when the range of the slider goes down for each expander that you add. For this reason I used a comparator with a resistor divider network (fig.7) for each gate input to have a high but predictable input impedance for each gate signal and then boost those signals up to the highest range available. Then I used zener based voltage reference diodes (fig.8, I used a lm4040CIM3-5.0 for the reference) to clamp the maximum voltage to as close to 5V as the references are rated for. In hindsight this was a pretty expensive solution that I will probably be changing in a future revision of Leaves to reduce cost but functionally it worked quite well. As it stands you can have at least 8 Leaves connected to the same Tree with no discernible drop in output voltage for the slider range, I haven't tested with more. While the voltage references are not necessary for every expander I would definitely suggest using the input comparators for any expander you might want to develop, they ensure that no matter how many other expanders are connect and whatever is going on with them that the gate signals being shared among them are accurate.

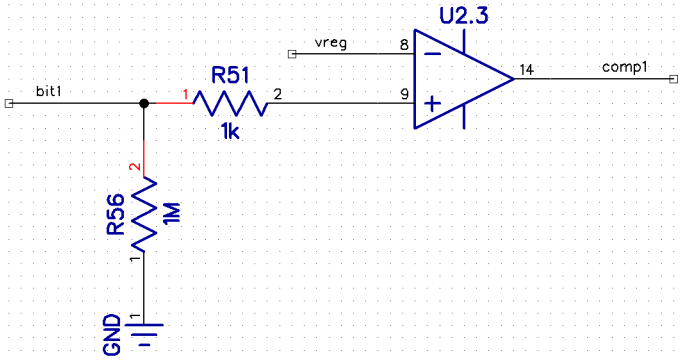

Fig. 7 – An input comparator acting as a buffer.

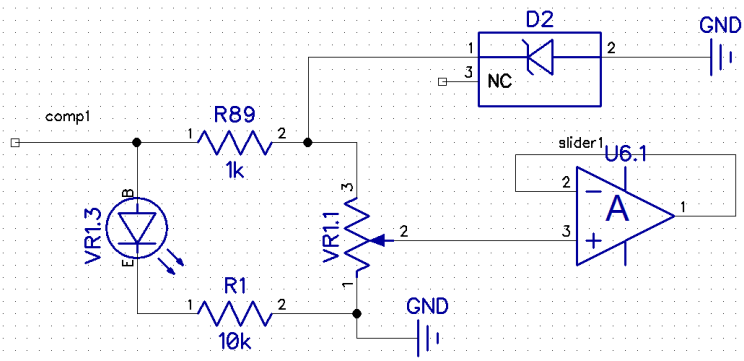

Fig. 8 – A voltage reference being used to feed the slide-pot

Overall I think there is a lot of untapped potential in the Turing Machine Expansion Protocol, it is an easy way to share 8 bits of data across a wide array of open source modules. I see a future where people are doing pretty advanced stuff like cellular automata or encoding the data using frequency modulation to make patching hosts and expanders modular in themselves. I've compiled what I have learned so far in here and will be adding as I make more expanders, if you are interested in making your own expander or have suggestions on how to improve this document please write me an email at: eli at mysticcircuits dot com.

References:

Doepfer Technical Details A-100:
https://doepfer.de/a100_man/a100t_e.htm

Random Sequencer Expander Documentation V1:
https://musicthing.co.uk/wp-content/uploads/2013/05/random-sequencer-expander-documentation-v1.pdf