TG-APP-ETRX2Power-100

ETRX2 WIRELESS MESH NETWORKING MODULE

APPLICATION NOTE – POWER CONSUMPTION

Telegesis

Table of Contents

1 Introduction

In order to calculate the average power consumption (or battery life) of an ETRX2 based product it is necessary to know the exact amount of power consumed for each individual activity.

In case the device is a router, or a coordinator this is straightforward as the unit will always consume 35.5mA in either transmit or receive mode (a router or coordinator is not supposed to go to sleep), but when it comes to end devices estimating the required power becomes more complicated.

This document gives an overview of common activities and the resulting power consumption to aid calculating the overall battery life.

Please note that in the same way that message transit times over multiple hops are non-deterministic in a mesh network, the power consumption may vary both with network utilisation and hop-count between source and destination of a message. Because of this the document at hand should only be regarded as a guideline. We highly encourage you to do in-system measurements for each individual application in a real life scenario to get the most accurate figures.

2 Power Consumption of Different Activities

For all experiments described in this chapter an ETRX2 is set to be a sleepy end device (SED) in power mode 2. The main difference from a router is that on an end device even in power mode 0 only the microcontroller is running and the radio only gets switched on when in use.

When not part of a network an ETRX2 will consume 9mA in power mode 0 because only the microcontroller is running. When scanning for PANs, trying to join a PAN or doing an energy scan, the radio is fully switched on and the unit will consume 36mA for the duration of the scan.

The device under test is attached to a Telegesis Development board and powered with 3.3V provided by the development board. The current consumption is measured in terms of a voltage drop across a 10Ω series resistor.

For the measurements three ETRX2 modules from different batches were used.

2.1 SED Polling -> No Message Present

For this measurement the built-in functionality 0011 is used to poll the parent at regular intervals, whilst it is made sure that there is no data to be polled from the parent. The length of the interval between any two polls has no effect on the amount of energy consumed per poll.

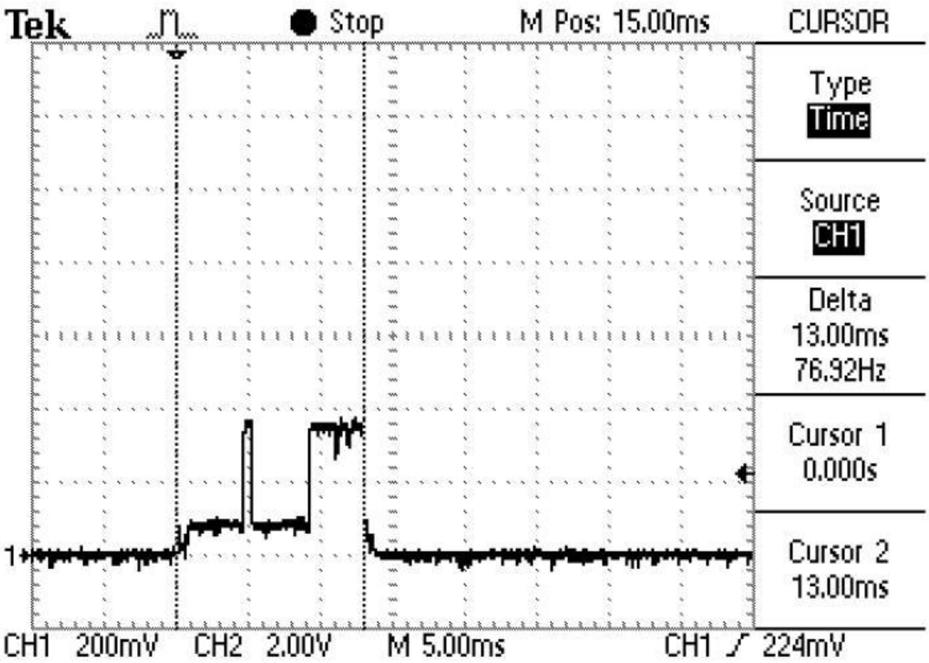


Figure 1: Polling

As seen in the figure above a poll consists 4 phases. In the first phase the on-chip microcontroller wakes up and the power consumption goes to about 9mA. After that the unit switches the radio on to send a poll, during which the power consumption goes up to 36mA (TX current). Next, the radio is switched off again to give the parent a chance of compiling a reply to the poll and finally the radio is switched back on to receive a response. Table 1 below gives an indication of how long the individual phases are and what the current consumption is.

Mode	Current Consumption	Time
Sleep	1uA	?
Wakeup	9mA	4.6ms
Send Poll	36mA	600us
Wait	9mA	4ms
Listen for reply	36mA	4ms
Sleep	1uA	?

Table 1: Poll for Data

The timings listed in the table above are averaged over 50 samples, as there can be slight variations of the consumption between samples. All samples were taken with no additional traffic on the air, additional traffic could potentially increase the power consumption (collision prevention, and actual collisions taking place).

Due to the internals of the stack it is to be expected that the measurements detailed above also apply to a mobile end device, given it has not moved away from the parent it has previously polled and it is polling within the timeout period.

Also polling by external interrupt in power modes 2 and 3 is expected to produce the same results, except that power mode 3 will save 0.5 to 1 microampere during sleep due to the internal timer being halted.

When averaging this data the average power consumption of a node depending on its poll time is:

Poll Interval	Average Consumption
0.5s	0.50mA
1s	0.25mA
2s	0.13mA
4s	0.06mA
8s	0.031mA
1 min	0.006mA

Table 2: Average Power consumption

Please note that when polling in intervals of less than 1 second only broadcasts, which are sent from the parent using just in time messaging, will be reliably buffered and passed on to the end device.

2.2 SED Polling -> Single Broadcast Message Present

As in the previous chapter the end device is polling for data. The only difference is that now there is a broadcast waiting to be passed on to the child. The payload of the broadcast is 30 bytes (plus 8 bytes containing the sender's EUI64).

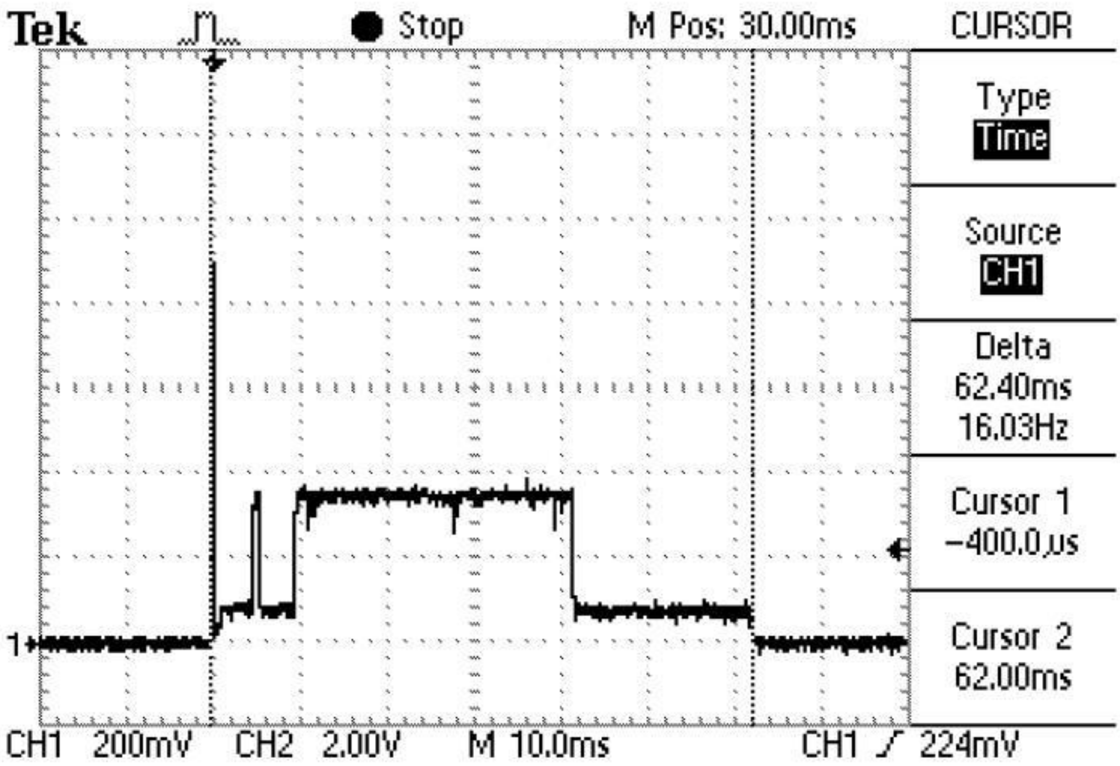


Figure 2: Poll for Data

In analogy to the previous chapter we can separate the polling into five phases. It can be seen that the only two differences between figure 1 and figure 2 are the length of the phase in which the stored data gets sent to the child using a just in time message from the parent and the following phase in which the microcontroller stays active to process the received message.

The updated timings for this scenario are shown in table 3.

Mode	Current Consumption	Time
Sleep	1uA	?
Wakeup	9mA	4.6ms
Send Poll	36mA	600us
Wait	9mA	4ms
Listen for reply	36mA	32ms
Processing Data	9mA	20ms
Sleep	1uA	?

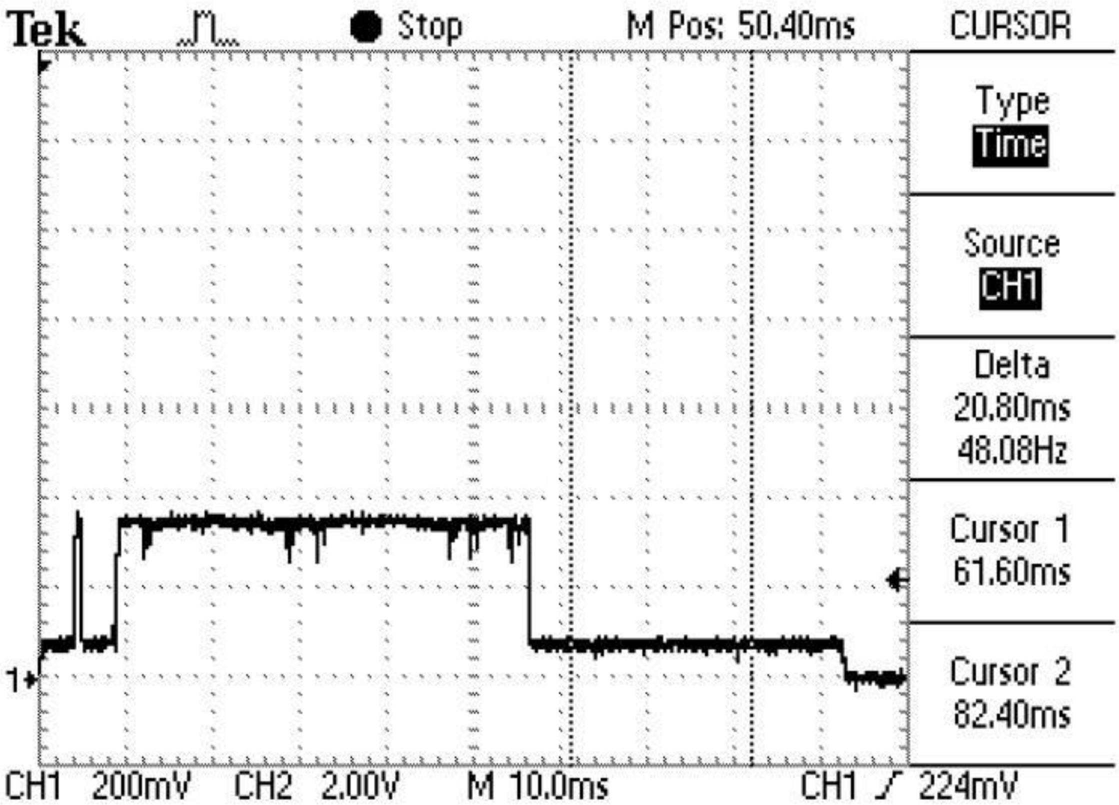
Table 3: Poll for Data

Due to the internals of the stack we expect to see the same timing in the case of a unicast stored on the parent, although in this case no just in time messaging is used.

Because unicasts do not use just in time messaging they are only stored for a limited amount of time, whereas broadcasts remain stored on a parent unless overwritten by a new broadcast.

2.3 SED Polling -> Two Broadcast Messages Present

Again, the end device is polling for data, but in this case there are two 30+8 byte broadcast messages stored on the parent, which are transmitted to the end device using just in time messaging.



Comparing these results to the previous chapter it can be seen that two of the sections are now taking longer, namely the one in which the radio is active in order to receive the response as well as the section in which the data is processed and prompted to the user.

Mode	Current Consumption	Time
Sleep	1uA	?
Wakeup	9mA	4.6ms
Send Poll	36mA	600us
Wait	9mA	4ms
Listen for reply	36mA	48ms
Processing Data	9mA	36ms
Sleep	1uA	?

Table 4: Poll for Data

2.4 SED Sending SCAST (single hop)

Now we are sending a scast to the (known) network's sink using functionality 0101.

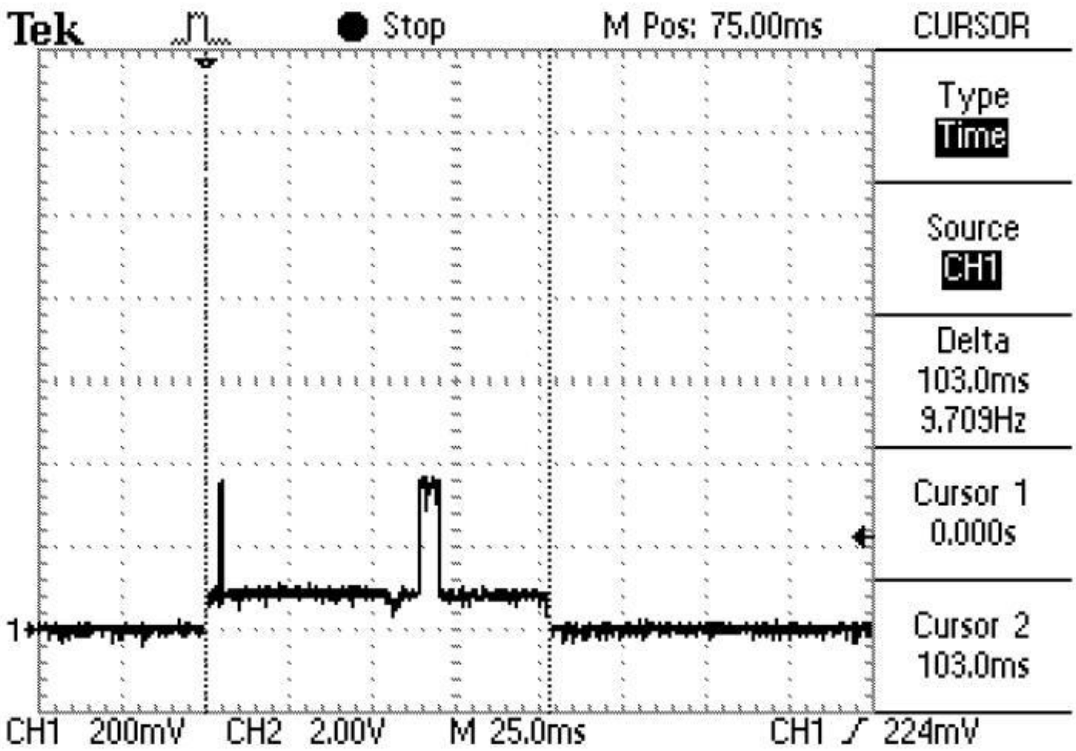


Figure 3: Sending a SCAST

It can be seen in figure 3 that this interaction between parent and child can be divided into 5 sections. In the first section the unit wakes up from sleep and assembles the packet, which gets sent in section 2. After sending the packet the node will insert a delay during which it is giving the parent time to evaluate the message and compile an acknowledgement (section 3). After about 50ms the node will poll for an ack and on reception do some tidy up work and go to sleep.

In order to get an end to end acknowledgement the node will continue to wake up and poll for the end to end acknowledgement every 500ms until timed out. Polling for an acknowledgement will look like as described in chapters 2.1 and 2.2. The more hops the message is travelling the more polls will be done until the acknowledgement is received, so the power consumption is increasing with the number of hops the message is travelling.

Mode	Current Consumption	Time
Sleep	1uA	?
Wakeup	9mA	4.6ms
Send Scast	36mA	600us
Wait	9mA	58ms
Listen for ack	36mA	7ms
Tidy Up	9mA	30ms
Sleep	1uA	?

Table 5: Send SCAST

For this scenario the Sink was also the parent of the SED, so the end to end acknowledgement was received with the next poll.

3 Conclusion

The power consumption of an end device can rely on a variety of factors, therefore whenever doing battery life calculations the actual power consumption should be measured in system to get real life data.

Factors do not have to be as obvious as payload lengths and poll rates, but also both the number of hops that end to end acknowledged messages are travelling and the overall network traffic (or in-band noise) can have an impact on the power consumption of a wireless mesh network.

In this document some guidelines are provided to give you a rough estimate to see if the project you are planning is feasible with a given power supply (battery).

Please note that in order to achieve ultra low power consumption it is required to either define all I/Os to be outputs, or to pull all inputs to a defined level as floating input pins will increase the current consumption. Using the build in pull-ups will also increase the overall current consumption. Furthermore as described in the hardware manual a pull-down of 10kΩ must be attached to the SIF_MOSI pin for lowest possible power consumption.

4 Trademarks

All trademarks, registered trademarks and products names are the sole property of their respective owners.

5 Disclaimer

Product and Company names and logos referenced may either be trademarks or registered trademarks of their respective companies. We reserve the right to make modifications and/or improvements without prior notification. All information is correct at time of issue. Telegesis (UK) Ltd does not convey any license under its patent rights or assume any responsibility for the use of the described product

6 Contact Information

Website: www.telegesis.com

E-mail sales@telegesis.com

Telegesis (UK) Limited

Marlow Business Centre

84 Station Road

Marlow

Bucks. SL7 1NX

UK

Tel: 01628 894347

Fax: 01628 894333

7 References

Telegesis - www.telegesis.com

Ember - www.ember.com