

PACKET STATUS REGISTER

President's Corner	01
TAPR at Hamvention	02
New Time and Frequency Products: TADD-2 and TADD-2 Mini	
Frequency Dividers	03
Treasurer's Report for 2008	05
A Cheap SDR Loran-C Frequency Receiver	09



President's Corner

BY STEVEN BIBLE, N7HPR, VICE PRESIDENT, TAPR

I'm writing the President's Corner this quarter while Dave is recuperating from surgery.

For those of you who know Dave, you know he won't be down long. We all wish him a speedy recovery. He did say he would be at Hamvention. See, nothing can keep him down!

What have we been up to this past quarter? TAPR's involvement with HPSDR remains strong. The LPU kit is available and can be ordered online. Sourcing parts for Alex has been challenging. We have been working with an off shore contract manufacturer to wind the many coils. It's a tedious process to make sure they are done correctly. This has taken more time than we thought. The Pandora enclosure is nearly ready for production and prototypes will be on display at the TAPR Hamvention booths (#455-458). Look for upcoming production announcements for

the Pennywhistle 20W PA and the Excalibur 10-MHz Frequency Reference.

The new TADD-2 frequency divider kit is now available. It accepts a 5 or 10 MHz input and provides six low impedance outputs. Each can be set to a rate of 1 PPS (with two duty cycles), 10 PPS, 100 PPS, 1000 PPS, or 10K PPS. For more information, go to page 3.

What are we up to next quarter? Hamvention is only days away. We have a great line up for the Friday morning TAPR Digital Forum starting at 9:15 AM in Room 1.

- Intro and TAPR Update - Steve Bible, N7HPR
- Timing Products Update - John Ackermann, N8UR
- HPSDR Manufacturing Update - Scotty Cowling, WA2DFI

- From Napkin to Ham Shack: Creating a TAPR Project - John Ackermann, N8UR

- Sharing Your Experience, Documenting Your Journey - Larry Wolfgang, WR1B

- ARRL/TAPR Digital Communications Conference Update - Steve Bible, N7HPR

We are pleased to announce that Richard Garriott, W5KWQ, will be the keynote speaker at the AMSAT/TAPR Banquet Friday night. Be sure and get your tickets in advance. The deadline is May 11th. Tickets will not be available at Hamvention. To order, see the announcement later in this newsletter.

Come see us at the TAPR booth and see what we are up to. Join in the fun, talk and enjoy the camaraderie.

- Steve, N7HPR

###

TAPR at Hamvention

By **STAN HORZEPA, WA1LOU**

TAPR will be at Hamvention in Dayton, Ohio, May 15-17.

TAPR's BOOTH SPACE

See TAPR at booths 455-458 (next to AMSAT's booths) in the Ball Arena, where we will have HPSSDR equipment up and running along with other TAPR goodies, like the new TADD-2 frequency divider kit.

TAPR DIGITAL FORUM

The TAPR Digital Forum will be 9:15 to 11:15 AM on Friday, May 15 in Room 1. Steve Bible, N7HPR, will moderate and the presentations will include:

“Introduction and TAPR Update” by Steve Bible, N7HPR

“Timing Products Update” by John Ackermann, N8UR

“HPSSDR Manufacturing Update” by Scott Cowling, WA2DFI

“From Napkin to Ham Shack: Creating a TAPR Project” by John Ackermann, N8UR

“Sharing Your Experience, Documenting Your Journey” by Larry Wolfgang, WR1B

“ARRL/TAPR Digital Communications

Conference Update” by Steve Bible, N7HPR

Immediately following, TAPR Secretary, Stan Horzeps, WA1LOU, moderates the APRS Forum at 11:30 AM to 12:30 PM, also in Room 1.

TAPR/AMSAT BANQUET

The third annual TAPR/AMSAT Banquet will be on Friday evening, May 15, 2008.

The banquet will be in the Jefferson Room at the Kohler Presidential Banquet Center, 4572 Presidential Way, Kettering, OH (just south of Dayton).

Richard Garriott, W5KWQ, will be the keynote speaker. Richard was aboard the ISS (International Space Station) in October 2008 and delighted many Amateur Radio operators and school children with contacts. He was the sixth private citizen to make the trip. In 1983, Richard's father Owen Garriott, W5LFL, was the first person to make Amateur Radio contacts from the Space Shuttle.

Reservations are required and must be made by Monday, May 11, 2009; tickets will **not** be sold at Hamvention. The price for the banquet is \$30 per person. Reserved tickets may be picked up at the AMSAT booth at Hamvention on Friday, or at the

door Friday evening.

Make reservations online in the AMSAT Store at <http://www.amsat.org> or phone Martha at the AMSAT office, 10 AM to 6 PM EDT at 301-589-6062 or toll free (in the US) at 888-322-6728.

BANQUET DETAILS

6:30 PM: Doors open and cash bar is available with beer, wine, liquor and soft drinks.

7:15 PM: Buffet dinner service begins.

The menu includes:

- Garden salad with assorted dressings
- Barbeque Baby Back Ribs Marinated
- Roasted Garlic Rosemary Chicken Breast in lemon butter sauce
- Salmon with Newburg sauce
- Petite Double Stuffed Baked Potatoes
- Normandy blended green beans
- Fresh fruit bowl
- Roll and butter
- Coffee/ iced tea/ water
- Assorted pies

New Time and Frequency Products:

TADD-2 and TADD-2 Mini Frequency Dividers

By JOHN ACKERMANN N8UR

Frequency counters are great tools. But they have some limitations: they aren't very good at describing changes in frequency over time, and in fact they aren't of any use at all in telling what time it is.

As a result, many time and frequency measurements rely on phase rather than frequency. Phase is just the difference in time between two "clocks." By measuring the rate at which one clock ticks versus the other, we can learn a lot about their relative performance, and that data can be converted back to frequency error if desired.

While phase can be measured at any frequency, it's often very convenient to use one pulse per second – abbreviated as a "PPS" signal. For example, GPS receivers used for timing generate a very precise PPS signal. While some clocks and frequency standards provide a PPS output, many don't. It's not too hard to wire together a bunch of divider chips to turn a 5 or 10 MHz signal into a pulse-per-second output, but some of the chips commonly used drift quite a bit with temperature, one-off IC projects can be clunky, and they usually don't have the right configuration for the next experiment you want to do.

When I designed the TADD-1 distribution amplifier,



TADD-2 FREQUENCY DIVIDER

my intention was to do a PPS divider project next, and to give it the flexibility a useful lab instrument should. However, I got bogged down in an overly complicated design using technology that was new to me (a CPLD chip), and at the time I needed a PPS distribution box for my existing signal sources more than a divider. So, the planned TADD-2 was put on the back burner, and the TADD-3 came out instead.

A number of years ago, one of the "time-nuts" gang, Tom Van Baak, wrote a very clever program for a PIC processor that acted as a single-chip frequency divider.

Clocked with a 10 MHz input, it would generate outputs at 1 PPS as well as several other frequencies, with very little jitter. It also had the ability to synchronize itself to an external pulse. However, Tom's code was limited to a 10 MHz input frequency; while that is very useful, a lot of frequency standards have a 5 MHz output. Some time later, another time-nut, Richard McCorkle, modified Tom's code to work at either 5 or 10 MHz. Tom and Richard both graciously agreed to Open Source their code, and a few months ago I started work on the long-delayed TADD-2, designed around a PIC divider. A follow on from that project, the TADD-2 Mini (T2-Mini) is also in the works.

TADD-2*

TAPR is now pleased to announce that the TADD-2 frequency divider kit is now available. Member price will be \$62; non-member price will be \$69.

* In this case, "TADD" stands for "Totally Awesome Digital Divider." As you may recall, for the TADD-1 it stands for "Totally Awesome Distribution Device" and for the TADD-3 it stands for "Totally Awesome Digital Distribution." Here at TAPR, we are doing our part to preserve our natural acronym resources.

The TADD-2 accepts a 5 or 10 MHz input (jumper selectable) and provides six low impedance outputs of about 2.75 volts into a 50-ohm load. Each output can be set to a rate of 1 PPS (with two duty cycles), 10 PPS, 100 PPS, 1000 PPS, or 10K PPS. The outputs are normally low, going high on the leading edge of the pulse. However, each can be inverted (i.e., normally high, going low on the leading pulse edge). There is also a seventh auxiliary PPS output available that is TTL level, 1 PPS only, and cannot be inverted.

The input circuit allows signal levels as low as -10 dBm to clock the divider. Like the TADD-1, the RF input is transformer coupled and DC isolated. A jumper allows either high impedance or a 50-ohm termination for the input.

A synchronization circuit allows the divider to be stopped, reset, and then synchronized with the leading edge of an incoming pulse. This allows the output pulse to be aligned with, for example, a GPS signal. Synchronization is a two-step process. Pressing the ARM switch causes the counter to stop, reset, and wait for the arrival of the SYNC pulse. After synchronization, the output will be within one clock cycle (e.g., 100 nanoseconds at 10 MHz) of the SYNC pulse. The SYNC pulse must be at TTL levels, but its polarity can be jumper selected.

The TADD-2 is the same size (roughly 6 x 4 inches) as the TADD-1 and TADD-3, and will fit into the TADD enclosure. It runs from a 12-volt supply (with onboard

voltage regulators) and draws from 50 to about 300 mA depending on the output load and polarity.

Preliminary measurements of the TADD-2's jitter (in other words, how much the PPS output wanders around due to circuit noise) shows a standard deviation of about 30-40 picoseconds, not much more than the noise of the best available time interval counters.

I mentioned above that phase shift over temperature could be a problem with dividers made out of cascades of ICs. I have done some very preliminary



TADD-2 MINI FREQUENCY DIVIDER

measurements on the TADD-2, and it seems to perform very well in this regard. Measuring from room temperature up to about 73 degrees C, the PPS output changed phase by about 3.25 nanoseconds, or 60 picoseconds per degree. I plan to rerun these measurements shortly with improved temperature monitoring to get more reliable results.

The TADD-2 uses through-hole components and will be available as a kit only. A PIC programmed with the divider code is included, and the source code is available for those who like to tinker.

TADD-2 MINI

In talking with Tom Van Baak about the TADD-2 design, we came up with the idea of a smaller version of the divider, a “dongle” that could easily be used for temporary experiments, or mounted inside an existing frequency standard. I decided to create a small, simple version of the divider. The end result is a circuit board that is 2.0 inches long by 0.75 inches high. It is built using surface mount components (except for the PIC chip itself, which is an 8 pin DIP package).

The TADD-2 Mini (or “T2-Mini”) is a much simpler device than the full-size TADD-2. It has only a single low impedance output channel, and provides only a PPS output (no higher frequencies). The output polarity isn't selectable. Changes in input frequency require changing solder jumpers rather than moving shorting blocks. On the other hand, it's tiny and it's cute.

The T2-Mini is still under development, but is progressing rapidly. Because it uses small surface mount components that may be difficult for some users to handle, TAPR will offer it as an assembled and tested unit only. The price has not yet been determined. TAPR hopes to start shipping the T2-Mini this summer.

Treasurer's Report for 2008

SUBMITTED BY TOM HOLMES, N8ZM, TREASURER

For the fiscal year ended 12/31/2008, TAPR's finances look to be in very good shape, but there are some caveats to keep in mind. The biggest one is that while we appear to have over \$135,000 in assets that is mainly because of our sales model, which takes in sales dollars before we pay for the parts that go into the HPSDR boards. In this case, Mercury sales just in the month of December totaled almost half of our total kit revenue for the year, but we had yet to pay for any of the parts or assembly labor that goes into it. The evidence for this is that our total December expenses for all categories amounted to less than one-third of the expenses for the year. When the Mercury bills are all paid, the books will show a more realistic picture of our finances.

This is not to imply that we are in dire straights, but merely to point out that the timing of this product distorts the picture a bit. A more normal picture of our situation would show us with about

\$40,000 combined from our checking and savings accounts, and the P&L situation to be enough better than break even to allow us to fund future development projects comfortably.

Given that the overall picture is healthy (kind of unusual in the current business climate, don't you think?), and that the interest lists for the LPU and the TADD-2 are running consistent with historic levels, TAPR will continue to be financially healthy through 2009.

The standard Balance Sheet and P&L reports for 2008 will be presented at the annual meeting at the DCC.

(Editor's Note: The Treasurer's Report is preliminary and subject to approval by the TAPR Board of Directors.)

###

WRITE EARLY AND WRITE OFTEN



Packet Status Register (PSR) is looking for a few good writers, particularly ham radio operators working on the digital side of our hobby, who would like to publicize their activities here.

You don't have to be Vonnegut to contribute to *PSR* and you don't have to use Microsoft Word to compose your thoughts. The *PSR* editorial staff can handle just about any text and graphic format, so don't be afraid to submit whatever you have to w1lou@tapr.org.

The deadline for the next issue of *PSR* is July 15, so write early and write often.

###

A Cheap SDR Loran-C frequency receiver

Poul-Henning Kamp

<phk@FreeBSD.org>
Den Andensidste Viking
Herluf Trollesvej 3
DK-4200 Slagelse
Denmark

ABSTRACT

Loran-C radio-navigation signals are broadcast in most of the northern hemisphere and offer a solid alternative to GPS disciplined frequency standards, but a regrettable lack of affordable receivers means that Loran-C sees very little use for this purpose.

This article describes a simple, cheap and efficient Loran-C frequency receiver for such purposes.

The three main components of the receiver is a homebuilt loop antenna, the TAPR/N8UR "ClockBlock" and the OliMex.com ADUC-P7026 microcontroller prototype card, for a total cost well under \$200.

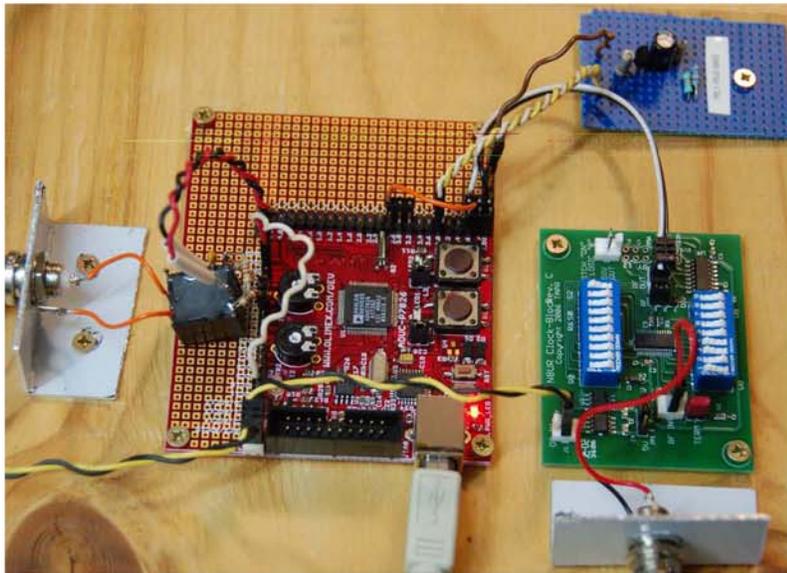


Figure 1. The complete receiver.

1. Introduction

LORAN-C signals, like GPS signals, offer wireless access to very stable and calibrated, and thus expensive, frequency sources and can therefore be used to steer cheaper oscillators to correct frequency.

Despite stern warnings to the contrary, the world today is more or less entirely frequency locked to the GPS satellites because that is the most convenient and cheap technology available.

This little project attempts to show that an equally cheap solution can be made based on the Loran-C signals, which are broadcast over most of the northern hemisphere.

Compared to GPS, Loran-C signals are just about as different as can be, where GPS satellites transmit their signals with only 50W using 1.2 GHz microwaves from 20,000 km above the Earth, Loran-C transmits are 200 meter tall steel towers which transmit several hundred kW at a frequency of 100 kHz.

This makes Loran-C the perfect backup for GPS, since there are almost no overlap between threats to the two systems.

This paper documents what is clearly a "work in progress", and as time and energy permits, I will update both this paper and the source-code that goes with it.

Poul-Henning Kamp

2. Hardware

The entire reason for this article is that a new breed of microcontroller combines a real 32bit CPU with fast analog/digital converters on a single chip.

Amongst these chips my fancy took to Analog Devices ADuC7026.

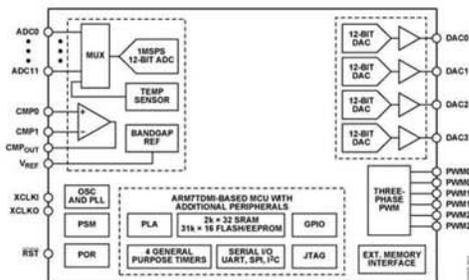


Figure 2. ADuC7026 block diagram

For one thing, Analog Devices are experts at the analog/digital border, and the specifications on the ADC in the ADUC is much tighter than on competing chips.

Furthermore the ADC offers a very convenient fully differential input mode and is not picky about the ADC reference voltage.

There are also some downsides: The chip needs a 42 MHz clock frequency to run the ADC at 1 megasample per second, and it only has 8kilo-bytes of RAM.

A very interesting feature is the built in "programmable logic array", which makes it possible to totally avoid external glue-logic in our case.

The main market for the ADuC7026 seems to be the variable frequency motor control market, and certain on-chip facilities are less than well thought out and documented.

One example of this is that the internal timers barely have any connection to the outside and that trigger conditions have unexplained systematic delays.

As it transpires, there is a way to hook things up in a way we can use.

2.1. The 42MHz clock frequency

The 42MHz clock frequency is much less of a problem than it sounds, thanks to John Ackermann, aka N8UR, we can simply pick up a "ClockBlock" from TAPR.org, set the jumpers correctly and get 42MHz out of almost any input

frequency we care to use.

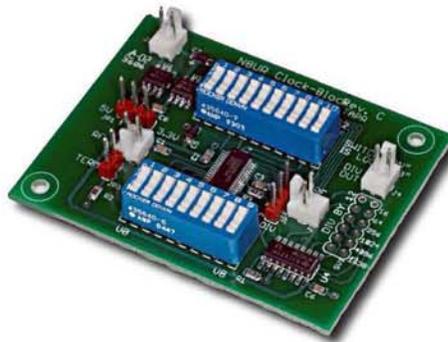


Figure 3. TAPR.org ClockBlock card

It is important to jumper the ClockBlock for 3.3V so it does not overdrive the ADuC7026 clock input.

The calculation of correct jumper settings is semi-complicated, and by far the easiest way is to use the web-page IDT provides:

<http://timing.idt.com/calculators/ics525inputForm.html>

For your convenience, here are three typical jumper settings:

$F_{in} =$	1MHz	5MHz	10MHz
S0	0	0	0
S1	0	0	0
S2	1	1	1
R0	0	0	0
R1	0	0	0
R2	0	0	0
R3	0	0	0
R4	0	0	0
R5	0	0	0
R6	0	0	0
V0	0	0	1
V1	1	1	0
V2	0	0	1
V3	1	0	1
V4	0	0	0
V5	0	1	0
V6	1	0	0
V7	1	0	0
V8	0	0	0

Figure 4. ClockBlock jumper settings

2.2. The ADuC7026 microcontroller

Being mostly of the software persuasion, I find the prospect of soldering 80 pin SMD packages something to be avoided if reasonably possible.

Fortunately my favourite Bulgarian electronics pusher, OliMex.com, has an ADUC-P7026 prototype card which is perfect for this project.

In the USA, you can purchase it from Sparkfun.com.

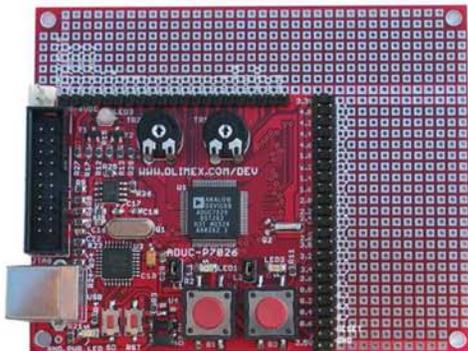


Figure 5. Olimex.com ADUC-P7026 card

2.3. The Antenna

I do not really consider the antenna part of the project, because I reused a very simple loop antenna I built for a previous round of Loran-C experiments, based loosely on a diagram I found on the website VLF.it, using an AD797 operational amplifier.



Figure 6. \$20 homebrew loop antenna

It goes without saying that I have not spent a lot of time on antennas, but roughly speaking, the requirements for the antenna input signal are something like:

- A Relatively flat passband from at least 70 to 130 kHz.
- Not too much signal above 300-400 kHz.

- At least 100mV peak-to-peak Loran-C signal.
- No more than 3V peak-to-peak total output signal.

2.4. The Antenna Balun

The ADC in the ADUC 7026 microcontroller can operate in a number of modes, but by far the most convenient for our purposes is the truly differential mode.

A differential CMOS ADC can be conveniently driven by a signal transformer of the kind found in old ADSL or ISDN equipment.

One particular nice thing about this approach is that a transformer has offset voltage.

ANTENNA BALUN

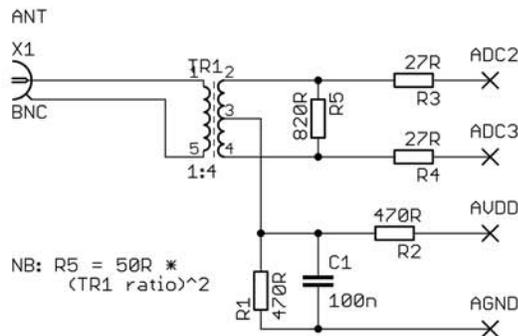


Figure 7. Antenna Balun Schematic

To be honest, this balun consists of whatever I found in my junkbox and is not optimized in any way shape or form.

The transformer I found is a 1:2 transformer from an ISDN NT box, where both sides have center taps. I feed only one half of the input side so it looks like a 1:4 with secondary center-tap.

Notice that R5 should be chosen as the input impedance multiplied by the square of the transformers ratio:

$$R5 = Z_{in} \cdot N^2$$

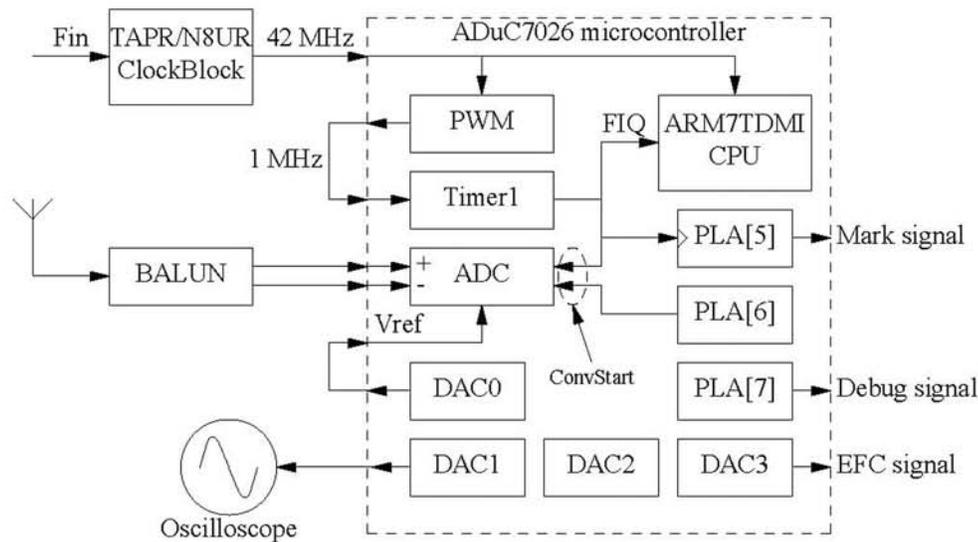


Figure 8. Blockdiagram of the receiver hardware.

2.5. Connecting it all

That is really all for the hardware, now we just need to connect it together:

- Jumper the ClockBlock for 3.3V supply.
- Connect the ClockBlock output to pin P0.7 on the ADUC-P7026.
- Connect your chosen frequency standard to the ClockBlock.
- Connect pin P2.7 ($PWMI_L$) to P0.6 (TI) on the ADUC-P7026 to establish a connection from the PWM output to the Timer1 input.
- Connect pin XXX to XXX on the ADUC-P7026 to establish a connection from DACXXX output to the ADC's VREF input.
- Connect the differential antenna signal to pin ADC1 and ADC2, remember that they both must stay between A_{gnd} and A_{vdd} at all times.
- Find a suitable power supply for both the cards, it should be at least 6VDC to avoid drawing power from the USB port, and needs to supply about 100mA total.
- Connect your computer to the USB port on the ADUC-P7206 card.

2.6. Available output signals

The following output signals are available for regulation and debug use.

- P2.1 ($PLAO[6]$) "Mark" output. The raising edge marks the start of measurements in the A-code GRI interval. The software reports how much later than this flank the 3rd zero crossing is found.
- P2.2 ($PLAO[7]$) "Debug" output. This pin flips low whenever and as long as the fast interrupt routine runs.
- P0.5 (ADC_{busy}) This signal is logic high while the ADC is performing a conversion
- DACXXX Analog replica of the averaged signal. Show this on your oscilloscope, use the "mark" output as timebase trigger.

2.7. Bootloader trigger circuit

The ADUC-7026 has a very convenient serial boot-loader which I use to download the firmware to the internal flash memory.

Unfortunately, there is no way to enter the boot-loader from software, it can only be triggered if P0.0 is logical low during a reset.

This little circuit implements a crude mono-table timer which can hold P0.0 down for us, while we trigger a reset from software, thus saving a trip down to the lab.

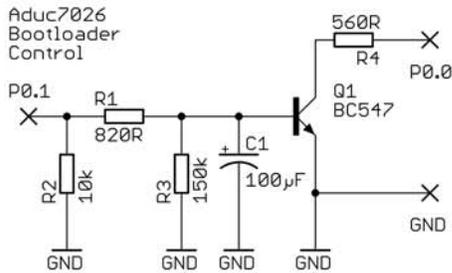


Figure 9. Bootstrap circuit

3. Loran-C signals

A Loran-C navigation chain consists of a master transmitter and from one to five slave transmitters, all broadcasting on the same "GRI", Group Repetition Interval.

A Physical transmitting station can participate in multiple chains, although for reasons of time-contention, typically it will only participate in two chains.

In our case, we are interested only in a single transmitter in a single chain, typically the nearest and strongest signal at the location of reception, although there is nothing preventing this construction from being used for more ambitious reception projects.

3.1. What the station transmits

There are many ways to describe the Loran-C signal, but for our analysis of how and why this receiver works, we will treat it as three components which are convoluted to give the actual on-air signal.

3.1.1. The GRI periodic function

This is basically a pulse generator which emits a pulse every $GRI * 10 \mu\text{seconds}$, so that for instance the Sylt chain with GRI 7499 has a period of 0.07499 seconds.

A very important feature of the GRIs used, is that they do not result in periods that are integral multiples of 1 msecond, so signals from CW radio transmitters will average out, allowing us to use essentially no other frequency domain filtering.

3.1.2. The signal code function

This is where much of the noise resistance of the Loran-C signal structure comes from: in alternating GRI periods two different pulse-trains are used for each of Master and Slave stations.

Whenever the GRI periodic function triggers a transmission eight or nine pulses will be transmitted with 1 msec interval, with polarity according to the following two figures.

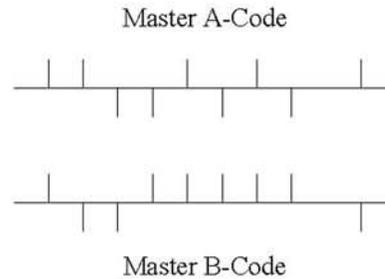


Figure 10. Master Signal Codes

In addition to the frequency filtering provided by the GRI averaging, the non-periodic nature of these codes provide significant improvements in S/N ratio.

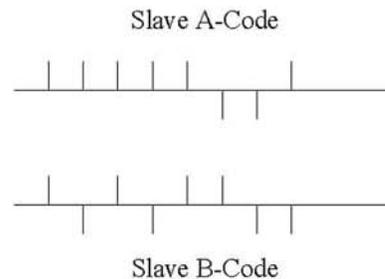


Figure 11. Slave Signal Codes

One important feature of both codes is that if the A and B interval codes are summed up, the odd numbered pulses cancel out leaving only four signals, spaced 2 msec apart.

3.1.3. The Radio Frequency Pulse function

The final step of the modulation is that each pulse is transmitted as

$$f(t) = \sin\left(\frac{2t\pi}{10}\right) t^2 \exp^{-2t/65}$$

where t has units of $\mu\text{seconds}$.

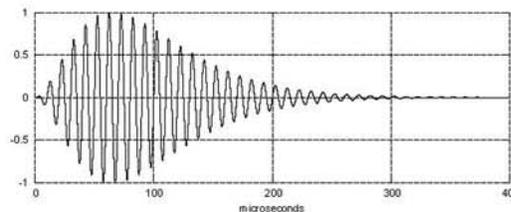


Figure 12. Loran pulse

In reality it is only the first part of the pulse-shape which is controlled, and in particular, the reference point is the 3rd positive zero-crossing of the signal

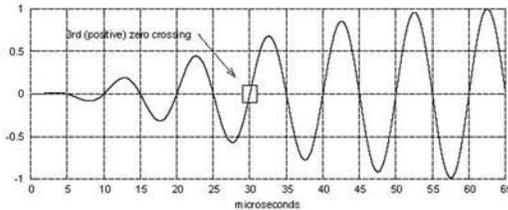


Figure 13. Loran pulse front

When we combine all these components, the result is something like this:

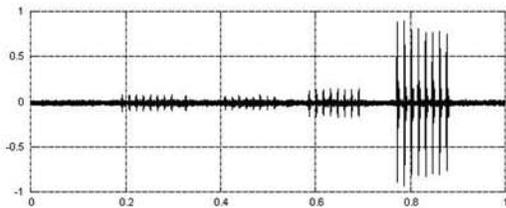


Figure 14. 6731 Lessay chain

All four stations in the Lessay chain are visible in this plot, first the master in Lessay, notice the "extra" 9th pulse 2 msec after the 8th, the follows the Xray slave in Soustons, the Yankee slave in Rugby and finally the Zulu slave on the island of Sylt.

Knowing the exact geographical locations of these four transmitters, and the time interval from the master transmission until the slaves transmit their pulses, we could calculate our geographic position from the measured time-intervals between these signals.

4. Reception

So, how do we find and lock onto the Loran-C signal with only 8 kilobytes of RAM ?

The brute-force method would be to simply average all samples over the FRI interval, that would allow us to identify all signals in the chain, tell A and B codes apart, locate the 3rd zero-crossing and track each signal.

Unfortunately, this would require up to $2 \cdot 9999 \cdot 10 = 199980$ averaging buckets.

Even if only one bit is used per bucket, this amounts to 24.5 kilobytes, more than three times the amount of RAM we have available.

We can get rid of the factor two by averaging only over the GRI interval, and provided we rotate every other GRI period one millisecond, we can still tell the four signal codes apart, at the expense of recognizing two energy levels in the pulse patterns.

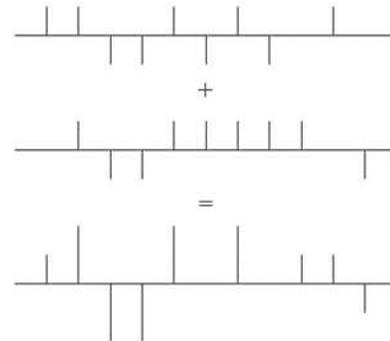


Figure 15. delayed master code average pattern

Unfortunately, that still requires more than 12 kilobytes of RAM.

It is possible to implement both of these versions of the algorithm by looking only at a small window of the GRI or FRI period at a time.

Since we realistically need 16, or preferably 32 bits, per bucket, and at best can use around 6 of the 8 kilobytes, this would result in at least 33 windows, in practice more because of necessary overlapping, each of which must be integrated for some time, before a signal may be apparent.

Instead we this narrow scan, single phase approach, we opt for a widescan, multi phase approach.

5. Phase Zero - Locating the strongest signal

Our first phase must locate the strongest signal in the chain, typically the transmitter closest to the receiving antenna.

When averaged over the GRI period, both the master and slave codes become four pulses separated by $2ms$ and our goal is simply to find the strongest one of these.

We do not care at this point, if it is a master or slave signal or where the A or B codes are in time, the next phase will figure that out.

Assuming we have 1000 buckets of 32 bits available, we can sample the GRI interval every $100\mu s$, and in doing so, we would likely miss even very strong signals: We need a sampling frequency which is not a divisor in the 100 kHz carrier frequency.

Since our goal is to just find the rough location of the strongest signal in the GRI, we are not even constrained to sample at rates that divide into the $1ms$ pulse spacing, and in fact get better results by not doing so.

Experiments and simulations has shown that if we average every $114\mu s$, every 114th sample, over the GRI period, we can expect to catch at least 30% of the peak amplitude of the strongest signal, and need only $99990\mu s / 114 = 887$ buckets.

5.1. Tek4014 view of phase zero

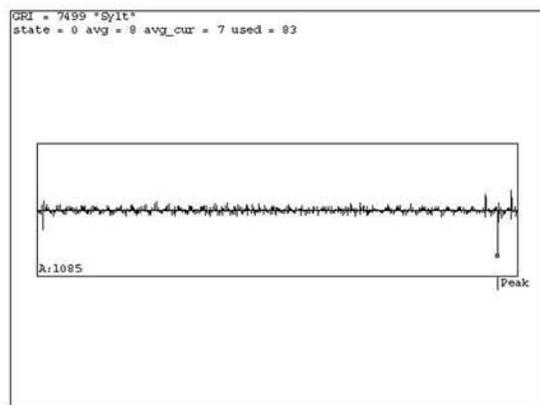


Figure 16. Tek4014 view of phase 0

This is how the tek4014 window looks in phase zero, we see the averaging buckets plotted, and the peak signal is marked.

In the lower left corner of the box is the amplitude of the signal in ADC units.

Notice that we do in fact quite clearly see all four peaks in the averaged signal, but one of them is significantly taller than the other three.

5.2. Hardware view of phase zero

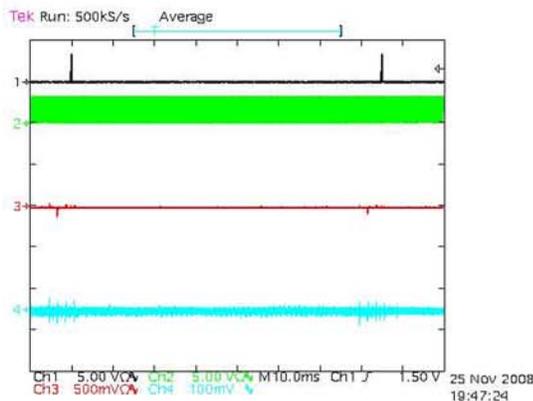


Figure 17. Phase 0 GRI wide signals

Trace 1 (black) is the marker output which is used to trigger the oscilloscope.

Trace 2 (green) is the debug output.

Trace 3 (red) is the DAC1 output, this is the same data which is plotted in the TEK4014 window.

Trace 4 (cyan) is the antenna input signal, where we can spot the LORAN pulses, thanks to a bit of averaging in the oscilloscope.

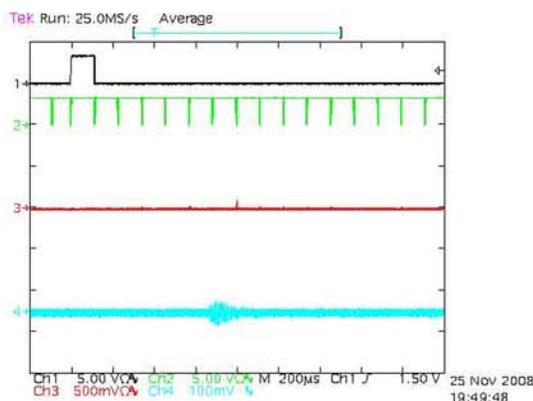


Figure 18. Phase 0 zoomed signals.

Zooming in, we see a single pulse of the loran signal in trace 4. Trace 2 shows the sample distance of 114 microseconds and trace 3 shows that we almost managed to miss this loran pulse entirely between two samples.

6. Phase One - Identifying the code

The peak signal from phase 1 must by definition be within 57 samples of the true peak value, of one of the four peaks, 2 msec apart, resulting from the GRI summing.

Now we need to find out if this is a master or slave station, and to identify which half of the FRI has the A and B code.

If we caught the last of the four peaks, the code starts 6 msec ahead of this point, if we caught the first of the four peaks, it spans another 9 msec after this point, so all in all, there are 16 windows we need to check for the presence of pulses.

We use a repetition frequency of 2 * GRI, sometimes called "FRI" for Frame Repetition Rate, in order to not add the A and B codes together.

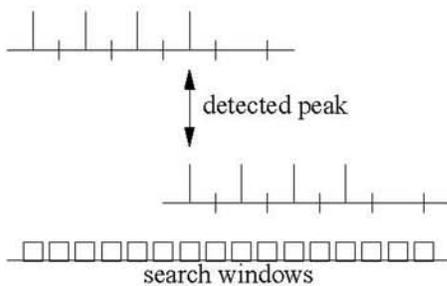


Figure 19. model code windows

Using no more than the 876 integers used in phase zero, each bucket can get 54 integers and if we space them 13 samples apart, each bucket will cover 702 μ seconds.

Again, 13 is a good number because it will make the sample points "wander" over the 100 kHz period of the pulses.

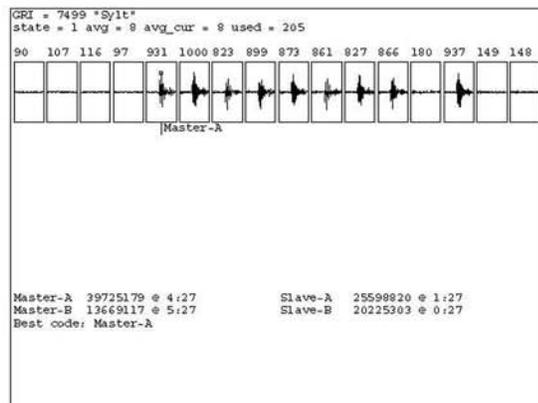


Figure 20. Tek4014 view of phase 1

The tek4014 view shows the 16 windows and shows the starting point of the strongest code in them.

Above each window is the energy content of that window, in thousands of the maximum found in any window. Due to timing effects, it is quite likely that one or two windows show numbers slightly above 1000. Presently, the energy levels is not used for code identification.

Below the windows are statistics for each of the best match for each of the four possible codes.

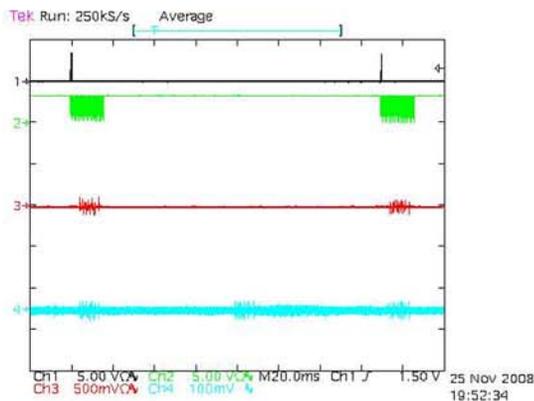


Figure 21. Phase 1 FRI wide signals

This is the hardware view, showing a full FRI interval.

Notice in trace 4, that we only sample every other pulse group.

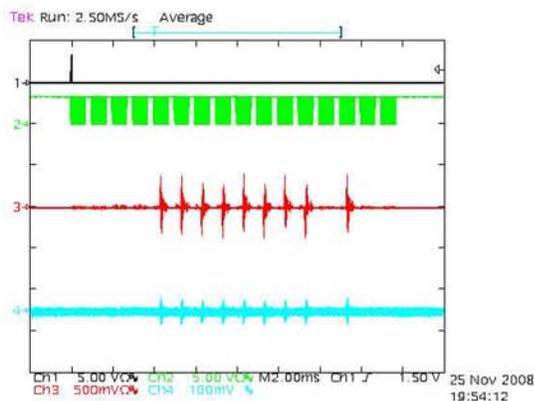


Figure 22. Phase 1 pulse group

Zooming in on the pulse group, we can see in trace 2 how each bucket is sampled and in trace 3 the resulting waveform.

Notice that the pulse shape is not fully recovered at this point, we only sample every 13

microseconds, but that is plenty to identify the code.

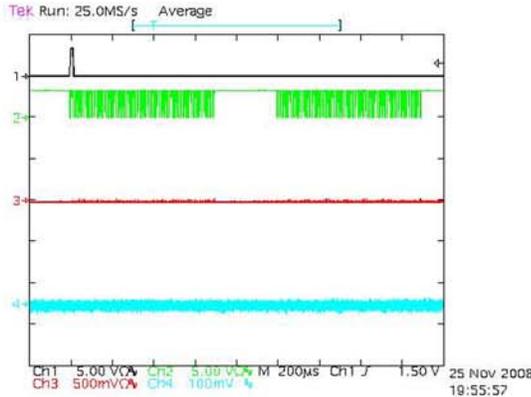


Figure 23. Phase 1 single pulse

Here we are zoomed into just two pulse windows.

7. Phase Two - Locking on to the signal

Now we have all the timing information we need to start integrating the signal for good: we know where the maximum is of the first pulse in the code, and we know which code it is.

In this phase we integrate every ADC conversion in a 750 (XXX: check code) sample window around all the 16 or 18 pulses in the signal, into the same single bucket, taking care to invert the pulses per the code.

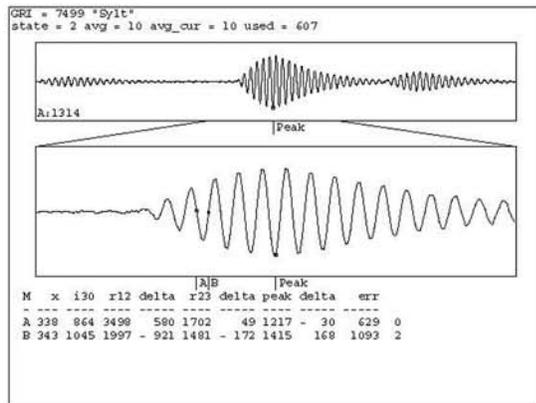


Figure 24. Tek4014 view of phase 2

The tek4014 view shows two views of our integration buffer, with the bottom one zoomed in on the peak value.

The various candidates for 3rd zero crossings are marked and their statistics given in numeric form below.

The MARK pulses are spaced 2 * GRI apart now, and we can see how the signal is sampled twice in that period.

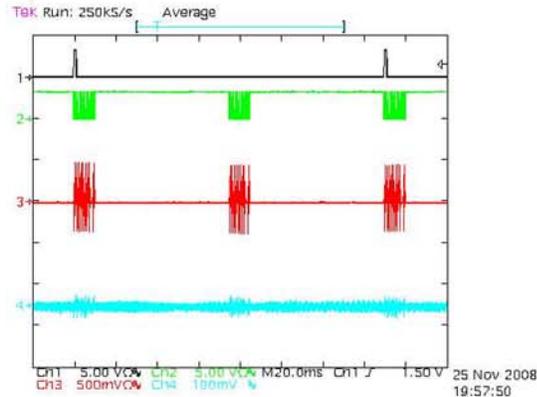


Figure 25. Phase 2 FRI view

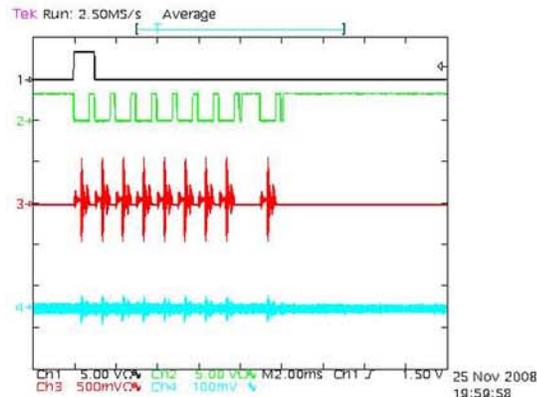


Figure 26. Phase 2 group view

Zooming in on the group, we can see the nine pulses of the master code being sampled.

Notice in trace 2, that there is a short burst of samples in the space between the 8th and 9th pulse, and after the 9th pulse. These samples are not technically necessary, but are planned for AGC use.

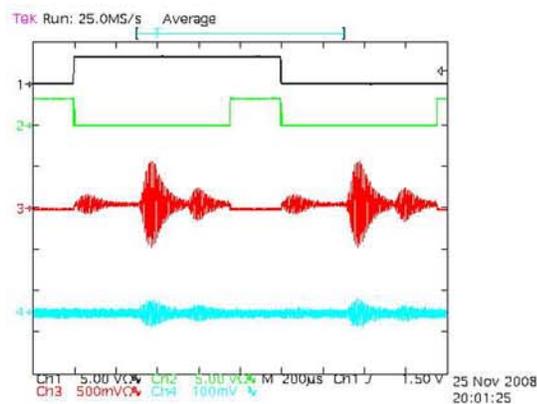


Figure 27. Phase 2 pulse view

Finally zooming in on a single pulse, we can see the full wave-shape of the Loran-C pulse, including the sky-wave echo.

The small "pre-echo" is currently unexplained.

7.1.

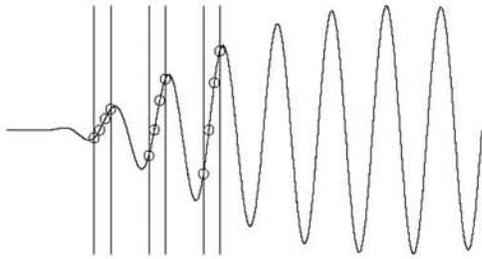


Figure 28. Finding 3rd zero

8. Phase Three - Tracking the 3rd zero-crossing.

In state 3 we have chosen the zero-crossing we want to track, and hold on to it to the best of our ability.

In practice we cannot simply nail a particular sample in the GRI window and track that, as the signal and the timebase may wander relative to each other.

We always use as tracking sample, the sample that is closest to the 3rd zero crossing.

Nominally the tracking point is sample 75 in the window, but we allow it to wander up to 10 samples in either direction, before we realign the sampling window to put it back at sample 75.

We interpolate the true zero crossing between the tracking sample and the two samples that surround it.

This is the Tek4014 display in phase three:

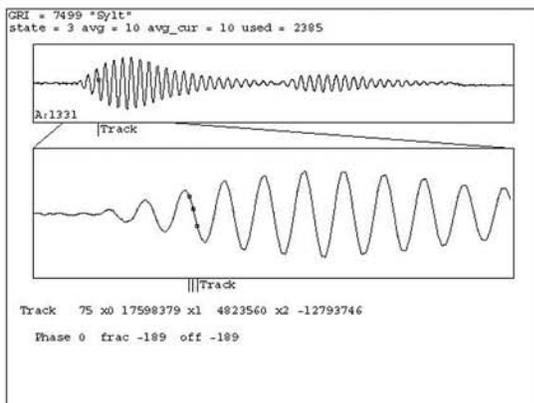


Figure 29. Tek4014 view of state 3

The top plot shows the entire sampling window and a marker for the tracking sample.

The bottom plot zooms in and marks both the tracking sample and the two neighboring samples.

Below the plots are the pertinent statistics, including calculated timing of the 3rd zero crossing.

The three samples which contribute to the interpolation of the zero crossing, are each filtered with a FIR bandpass filter:

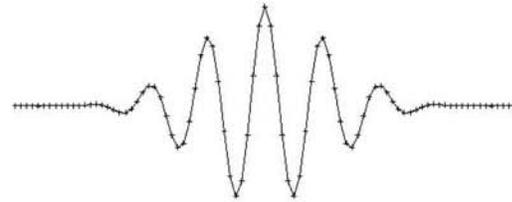


Figure 30. The FIR filter kernel

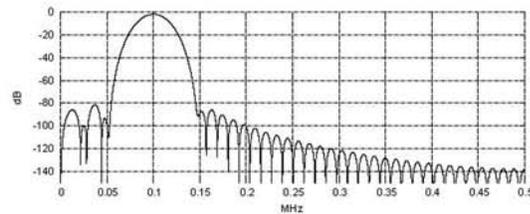


Figure 31. The FIR filter frequency response

The exact choice of filter has proven to be remarkably unimportant, but it does in fact reduce the noise in the three sample points considerably.

8.1. Interpolating the zero crossing

Since we want better resolution than the whole microseconds the sample rate provides, we need to interpolate the true zero crossing from the sample values around it:

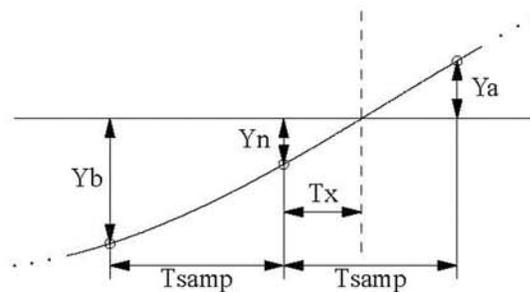


Figure 32. Interpolating the zero crossing

If we define Y_n as the sample that has the lowest absolute value, closest to the 3rd zero-crossing, we can determine the range of T_x by solving:

$$Loran(x) = -Loran(x + 1\mu s) \quad x \in [29\mu s \dots 30\mu s]$$

Doing so, we find $x \approx 29.664\mu s$.

It follows that the Y_n sample must be located between $[29.664\mu s \dots 30.664\mu s]$ and consequently that $T_x \in [-337ns \dots 664ns]$.

The analytical solution for finding T_x given Y_b , Y_n and Y_a is neither practically realizable with the limited amount of CPU and memory we have available, nor necessary.

Instead we calculate the FIR filtered signal at the sample nearest the zero-crossing (F_n), and the two samples right before (F_b) and right after (F_a) and find the ratio:

$$\frac{F_n}{F_a - F_b} \in [-0.2871 \dots 0.2679]$$

Which we map to the corresponding:

$$T_x \rightarrow [-337ns \dots 664ns]$$

Using a lookup table containing the precomputed conversion function:

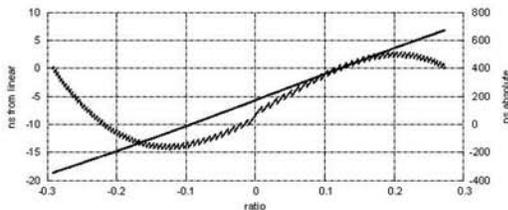


Figure 33. Interpolation non-linearity

The lookup table is built by the script `pulse_param.tcl` and lives in the source file `pulse_param.c`.

The script automatically reads in the FIR filter coefficients from the `fir.c` source file and determines how many entries the lookup table needs to have, to provide the specified resolution.

The table is extended 5% further than necessary in both end, to cater for edge effects and noise components.

For a resolution of nanoseconds, the table has 1111 entries.

8.2. Serial output data

agc	1324
avg	10
track	75
yb	22376693
yn	8627207
ya	- 8645207
Fyb	12611456
Fyn	2030392
Fya	-10263240
μs	105
Phase	0
frac	-185
off	- 185
#	

Figure 34. Serial data stream

9. Performance

These are some of the first performance data collected, and they should therefore be taken with all sensible precaution.

The signal is 7499M, 205 km away.

The exponential averaging factor is 1/1024.

The timebase is a free-running FRS10 Rb, the samples have been compensated for the randomly chosen, but deliberate 7.225e-10 frequency offset.

A phase measurement is recorded once per second using Timer2, from approx 2008-11-30 11:00 to 2008-12-01 11:00 UTC.

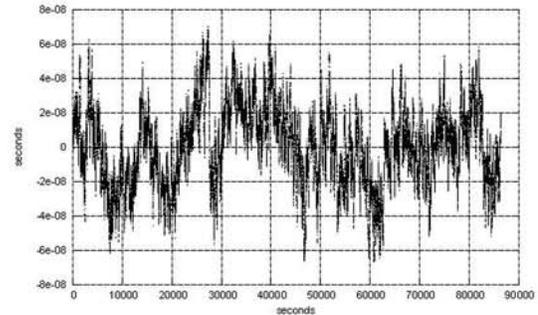


Figure 35. Raw offset samples

The standard deviation of the raw samples is 22.6 nanoseconds.

A histogram of the samples show a gaussian distribution, which seems well suited to more aggressive averaging.

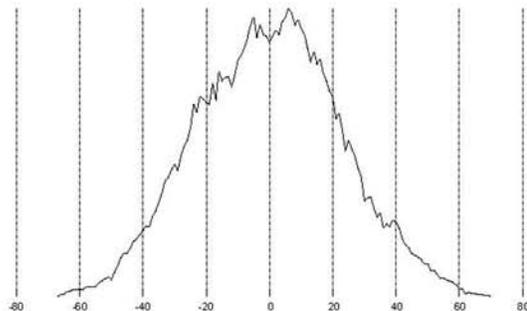


Figure 36. Offset value interval histogram

The modified allan variance is plotted below:

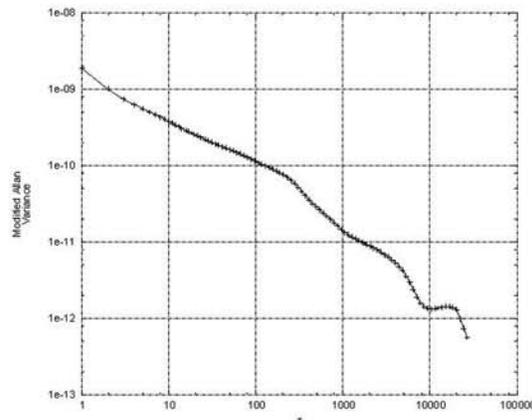


Figure 37. Modified allan Variance

10. Software

The software that goes with this article is written as proof of concept, but is written such that experimentation and further development is possible.

10.1. The tricky assembler bits

The tricky bit of the software is the assembler coded fast interrupt routine in the file *crt0.S*.

I have written this code so that it does not encode any of the high-level logic, but instead mechanically walks a chain of structures that tell it what to measure and when.

For instance, when we scan the 6731 GRI in phase zero, the specification looks like this:

```
{
    mark           = 1;
    ptr            = ss->ptr;
    polarity       = "+";
    avg            = 6;
    cnt            = 1;
    timer_repeat   = 114;
    repeat         = 590;
    timer_after    = 50;
    next           = this;
}
```

Figure 38. Phase zero specification

This specification tells the assembler code to pulse the MARK pin high when starting this specification, then 590 times take one ADC measurement 114 microseconds apart, then skip 50 microseconds and start over.

Since $590 * 114 + 50 = 67310$ this will repeatedly scan the 6731 GRI signals.

The *polarity* member specifies that the samples all have positive phase, the *avg* specifies the exponential average time constant and the *next* member tells it what to do next (the same thing).

With none of few modifications, this scheme should also support reception of multiple stations in the same chain or even several stations from different chains.

10.2. The high level logic

The highlevel logic, such as it is, is written in C code in the *loran0.c* file.

Right now the code is semi-manual, controlled by single characters received on the serial/USB port, running at 115200 bps.

The following commands are recognized:

- c This will present a menu with Loran-C chains from which you can choose one by entering the character in the [...].

```
-----
[@] 5543 Calcutta
[A] 5930 Canadian East Coast
[B] 5980 Russian-American
[C] 5990 Canadian West Coast
[D] 6042 Bombay
[E] 6731 Lessay
[F] 6780 China South Sea
[G] 7001 Bø
[H] 7030 Saudi Arabia South
[I] 7270 Newfoundland East Coast
[J] 7430 China North Sea
[K] 7499 Sylt
[L] 7950 Eastern Russia Chayka
[M] 7960 Gulf of Alaska
[N] 7980 Southeast U.S.
[O] 7990 Mediterranean Sea
[P] 8000 Western Russia
[Q] 8290 North Central U.S.
[R] 8390 China East Sea
[S] 8830 Saudi Arabia North
[T] 8930 North West Pacific
[U] 8970 Great Lakes
[V] 9007 Eiði
[W] 9610 South Central U.S.
[X] 9930 East Asia
[Y] 9940 U.S. West Coast
[Z] 9960 Northeast US
[ ] 9990 North Pacific
-----
```

Please select chain:

Figure 39. Chain selection menu

- t Make TEK4014 plot via serial/USB port
- u Zoom in on curves in TEK4014 plots
- d Zoom out on curves in TEK4014 plots
- a Double the exponential average time constant.
- b Half the exponential average time constant.
- 1 Skip to mode 1
- 2 Skip to mode 2

10.3. TEK4014 plotting

Before windows and mice, Tektronix produced a series of high quality graphical terminals, based on the storage-CRT principle.

These plotting instructions to these terminals became the de-facto format for plotting files and as a result, the original X11 "xterm" program offered, and still does, TEK4014 emulation.

If you connect to the serial/USB port using the X.org xterm program and press 't', then you will be rewarded with a nice plot as seen earlier in this paper.

If you use another terminal program which does not support TEK4014 plotting commands, you will get a blast of random ASCII characters.

10.4. Utility functions etc.

The other three C language files contains the code to configure the hardware, code to use the serial/USB port and a function which calculates

basic statistics for an array of integers.

The files `divdi3.c`, `qdivrem.c` and `divsi3.S`, are runtime support files for the GCC compiler.

10.5. Downloading firmware to the ADuC7026

You can either use a JTAG gadget or download the firmware via the serial/USB port.

Included in the source-code package is an "aduc" program I have written for that purpose.

10.6. Compiling the firmware

Cross-compiling is notoriously tricky, but the FreeBSD build environment makes it quite easy.

To build a arm cross-compiler + toolchain, simply do the following:

```
cd /usr/src
make toolchain \
    TARGET=arm TARGET_ARCH=arm
```

Figure 40. Building an ARM toolchain on FreeBSD

The Makefile knows how to find these tools under the `/usr/obj`.

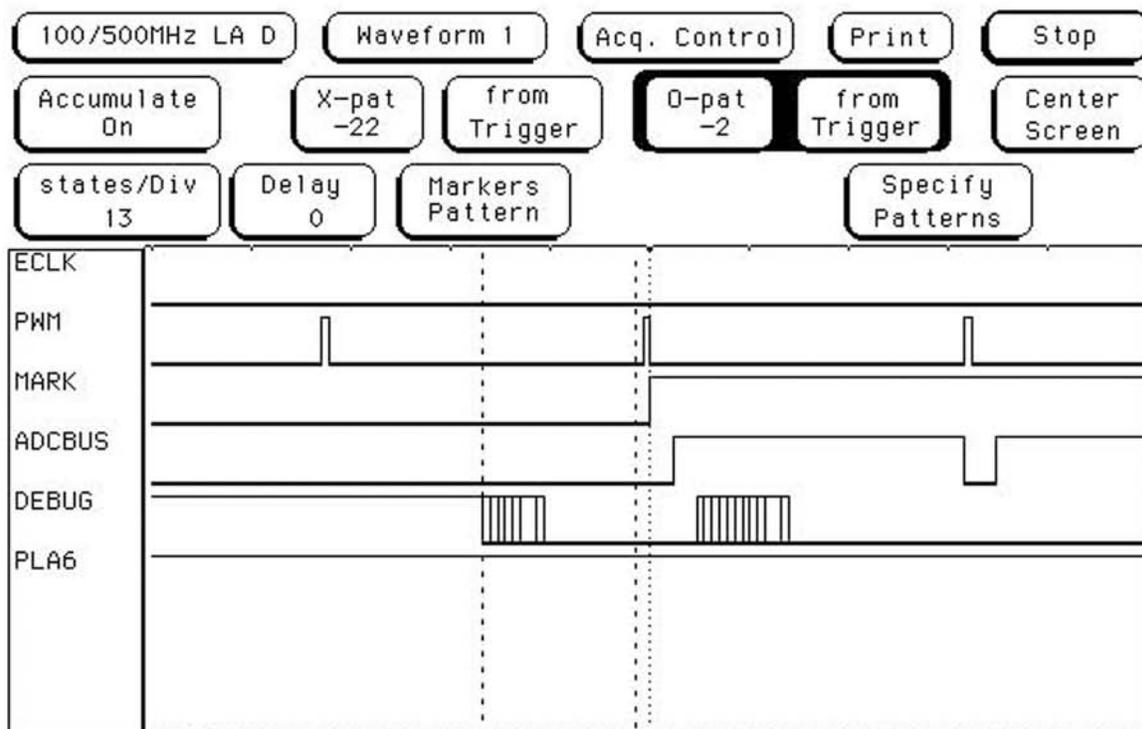


Figure 41. Protocol analyser trace

11. Timer1 unexplained

This is a confession: I have no idea exactly how Timer1 is supposed to work in the ADuC7026 chip, and the way I have observed it work baffles me, but the software manages to work around it.

The PWM section provides us with a 1MHz frequency which, through an external pin, becomes Timer1's clock signal.

In Fig XXX above this external signal is named PWM, which for this experiment, was give a very short "on" time.

The signal named "MARK" is from PLA[5], which is configured so the flip-flop is latched by Timer1's count-complete signal.

The signal named "ADCBUS" is the ADC busy signal, which indicates that the ADC has started performing a conversion, and the ADC is also triggered by Timer1's count-complete signal.

So far, so good, it looks like Timer1 expires around the middle of the figure.

But now look at the "DEBUG" signal, which is programatically lowered as the first thing in the fast interrupt handler, then pulsed high after Timer1 has been reloaded.

The fast interrupt is also triggered by Timer1's count-complete signal, but this happens well before PLA[5] and the ADC receive the signal.

From the FIQ request is raised until the first instruction of the handler is executed takes at least five clock cycles, but it can take as much as 13 (XXX ?) cycles, depending which instruction the CPU was executing.

This is why the "DEBUG" trace shows multiple transitions: it is not deterministic in time.

But the mystery gets deeper here, because the previous PWM signal, presumably the one that counts down to one, is a fair bit further ahead of the DEBUG signal than 5-13 (XXX) clock cycles.

But it gets more mysterious still: The fast interrupt handler loads a value into the T1LD register right before the high pulse on the DEBUG signal (right of the dotted line).

If the leftmost PWM signal is the 1 count, then the central one is the 0 count, it would be reasonable to expect that loading N or N-1 into the T1LD register at this point, would cause the next Timer1 count complete event to happen N cycles

of the PWM signal later, starting with the one in the right hand side of the trace.

Experimentally I have found that I have to load N-3 to make that happen.

For this to make sense, the PWM edge that triggered the count down to zero must be just outside the papers left edge so that the central PWM pulse is number 3 in the new counting cycle.

If that is the case, then the chip applies approximately $3\mu\text{s}$ worth of metastability latching on Timer1's count-complete output, before it reaches the ARM7TDMI core and the ADC unit.

At the 42MHz clock, 3μ correspond pretty precisely to 128 clock cycles.

If that is the depth of the latch-line that resolves metastability, then I would presume to think that it is sufficient deep.

Enquiring minds wants to know...

###

DCC DVDs For Sale

Did you miss the ARRL and TAPR Digital Communications Conference last September? Did you attend, and now want a record of some, or all of the talks?

Good news! ARVN:Amateur Radio//Video News captured every minute of the presentations on video, and now offers them on six DVDs in high quality video and audio.

The presentations are grouped on the six DVDs in as logical an order as possible. You can buy the DVDs individually (\$15/each, plus \$3 shipping, \$4 international). Or get whole set for \$75 (plus \$3/\$4 shipping). The complete set saves you \$30 over buying them individually, after shipping cost. You can order by credit card or PayPal at www.arvideonews.com.

###

Packet Status Register

#108 Spring 2009, ISSN: 1052-3626

Published by

TAPR

phone 972-671-TAPR (8277)

e-mail taproffice@tapr.org

URL www.tapr.org

TAPR Office Hours

Monday – Friday, 9 AM – 5 PM Central Time

Entire Contents Copyright © 2009 by TAPR. Unless otherwise indicated, explicit permission is granted to reproduce any materials appearing herein for non-commercial Amateur Radio publications providing that credit is given to both the author and TAPR, along with the TAPR phone number – 972-671-TAPR (8277). Other reproduction is prohibited without written permission from TAPR

Opinions expressed are those of the authors and not necessarily those of TAPR, the TAPR Board of Directors, TAPR Officers, or the Editor. Acceptance of advertising does not constitute endorsement by TAPR, of the products advertised.

Postmaster: Send address changes to TAPR, P. O. Box 852754, Richardson, TX 75085-2754. *Packet Status Register* is published quarterly by TAPR. Membership in TAPR, which supports the electronic publication of the *Packet Status Register*, is \$25.00 per year payable in US funds.

TAPR is a community that provides leadership and resources to radio amateurs for the purpose of advancing the radio art.

Submission Guidelines

TAPR is always interested in receiving information and articles for publication. If you have an idea for an article you would like to see, or you or someone you know is doing something that would interest TAPR, please contact the editor (wallou@tapr.org) so that your work can be shared with the Amateur Radio community. If you feel uncomfortable or otherwise unable to write an article yourself, please contact the editor for assistance. Preferred format for articles is plain ASCII text (Microsoft Word is acceptable). Preferred graphic formats are PS/EPS/TIFF (diagrams, black and white photographs), or TIFF/JPEG/GIF (color photographs). Please submit graphics at a minimum of 300 DPI.

Production / Distribution:

Packet Status Register is exported as Adobe *Acrobat* version 5 and distributed electronically at www.tapr.org

PSR Packet Status Register Editor:

Stan Horzepa, WA1LOU

One Glen Avenue, Wolcott, CT 06716-1442 USA

phone 203-879-1348

e-mail wallou@tapr.org

TAPR Officers:

President: David Toth, VE3GYQ, ve3gyq@tapr.org

Vice President: Steve Bible, N7HPR, n7hpr@tapr.org

Secretary: Stan Horzepa, WA1LOU, wallou@tapr.org

Treasurer: Tom Holmes, N8ZM, n8zm@tapr.org

TAPR Board of Directors:

Board Member; Call Sign, Term Expires, e-mail address

John Ackermann, N8UR, 2010, n8ur@tapr.org

Steve Bible, N7HPR, 2011, n7hpr@tapr.org

Scott Cowling, WA2DFI, 2009, wa2dfi@tapr.org

Eric Ellison, AA4SW, 2009, aa4sw@tapr.org

Stan Horzepa, WA1LOU, 2011, wallou@tapr.org

John Koster, W9DDD, 2009, w9ddd@tapr.org

Darryl Smith, VK2TDS, 2011, vk2tds@tapr.org

David Toth, VE3GYQ, 2010, ve3gyq@tapr.org

Bill Vodall, WA7NWP, 2010, wa7nwp@tapr.org

TAPR is a not-for-profit scientific research and development corporation [Section 501(c)(3) of the US tax code]. Contributions are deductible to the extent allowed by US tax laws. TAPR is chartered in the State of Arizona for the purpose of designing and developing new systems for digital radio communication in the Amateur Radio Service, and for disseminating information required, during, and obtained from such research.