



(<https://microship.com/>)

Art without engineering is dreaming;
Engineering without art is calculating.

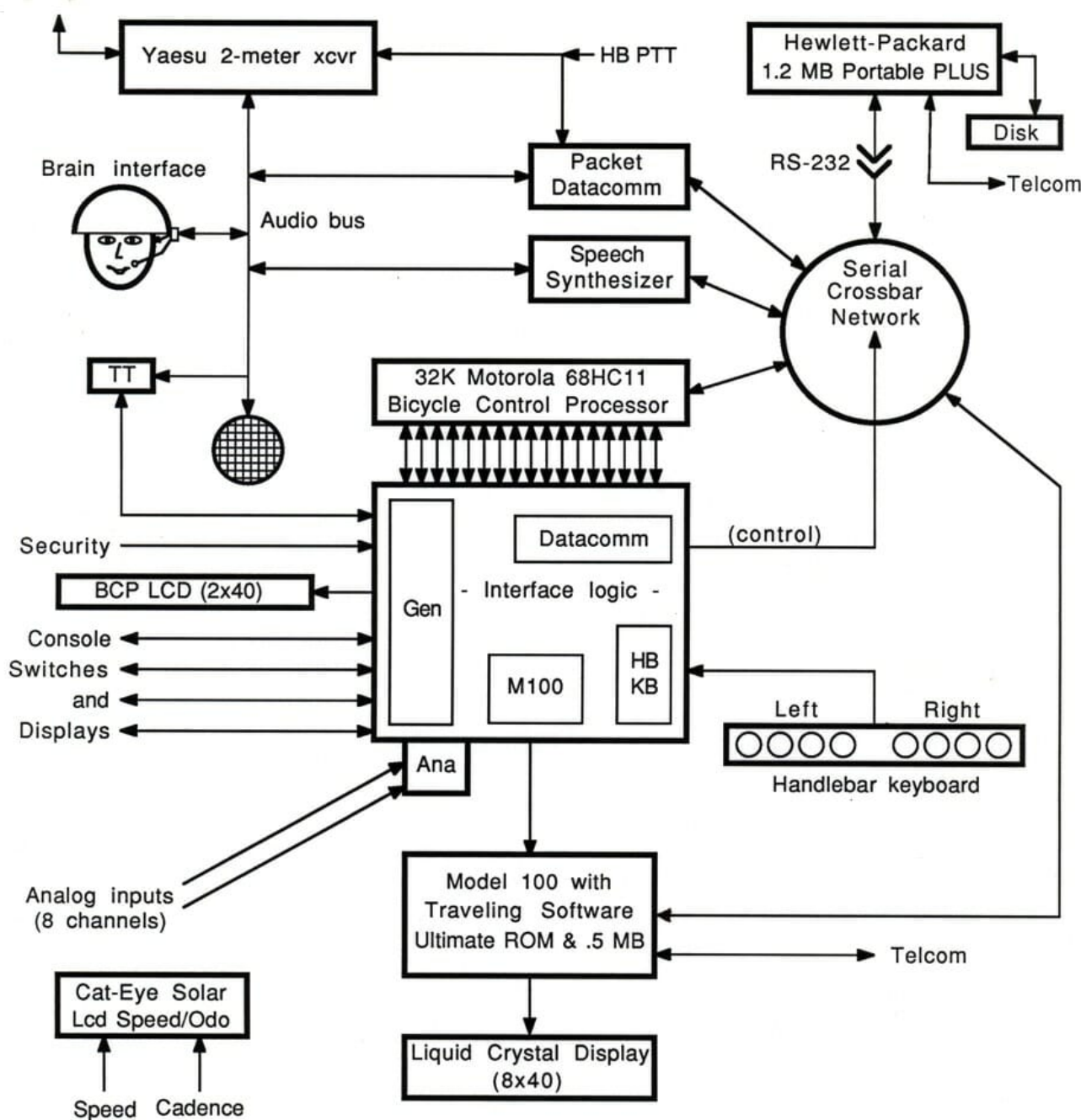
Steven K. Roberts, N4RVE



[s://w](#) [s://tw](#) [s://w](#)
[ww.fa](#) [itter.c](#) [ww.y](#)
[cebo](#) [om](#) [outub](#)
[ok.co](#) [/nom](#) [e.co](#)
[m](#) [adne](#) [m](#)
[/data](#) [ss\)](#) [/user](#)
[wake](#) [/micr](#)
[\)](#) [oship](#)
[\)](#)

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

Ok No [Read more \(https://microship.com/privacy-policy/\)](https://microship.com/privacy-policy/)



Winnebiko Control System Internal Architecture

Bicycle-Mobile Packeteering

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to our use of cookies. [View our privacy policy](https://microship.com/privacy-policy/).
By Steve (https://microship.com/author/steve-achon/) April 1, 1988

Ok No Read more (https://microship.com/privacy-policy/)

This has always been one of my favorites from the 73 Magazine series I wrote in the late '80s.

The editors gave me freedom to talk about everything from cultural issues to deep geekery, and I was creating a lifestyle that melded those in all sorts of twisted ways. In this one, the world-changing implications of mobile data communication are spotlighted... along with the tools I used to write while pedaling the bike.

For more on the system architecture, there is a piece about the Anatomy of the Winnebiko II (<https://microship.com/anatomy-of-the-winnebiko-ii/>). But this piece on "Packeteering" hints at the cultural changes that lay ahead, then describes the chord keyboard in detail.

It's time to pull packet radio out of its infancy!

by Steven K. Roberts, N4RVE

73 Magazine

April, 1988

My ramblings hardly seem like a bicycle trip. It hardly even seems like TRAVEL sometimes, with the familiar intertwined layers of Dataspace swirling around me no matter where I roam. Whether clipped into a modular jack in Jackson, hunkered down at a pay phone in Phoneton, or prowling through NET/ROMs from Netcong, I always feel a sense of connection with my stable electronic community. Somehow, the pedaling interludes seem but playful diversions — an endless source of tales and insights to keep those keyboard fingers exercised while living comfortably in Dataspace.

The network world is home alright, but culturally, the amateur packet community is still in its infancy. For well over a decade, a complex society has been developing in the large computer networks, prowled not only by the original hackers but by a diverse cross-section of humanity.

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

Ok No [Read more \(https://microship.com/privacy-policy/\)](https://microship.com/privacy-policy/)

Bicycle-Mobile Packeteering

It's time to pull packet radio out of its infancy!

by Steven K. Roberts N4RVE

My ramblings hardly seem like a bicycle trip. It hardly even seems like TRAVEL sometimes, with the familiar intertwined layers of Dataspace swirling around me no matter where I roam. Whether clipped into a modular jack in Jackson, hunkered down at a pay phone in Phoneton, or prowling through NET/ROMs from Netcong, I always feel a sense of connection with my stable electronic community. Somehow, the pedaling interludes seem but playful diversions—an endless source of tales and insights to keep those keyboard fingers exercised while living comfortably in Dataspace.

The network world is home alright, but culturally, the amateur packet community is still in its infancy. For well over a decade, a complex society has been developing in the large computer networks, prowled not only by the original hackers but by a diverse cross-section of humanity.

I mention all this because hams possess a unique set of tools quite the envy of traditional networkers everywhere. Yet hams have hardly begun to develop the "nth-layer protocols" that govern style and behavior... the essential underpinnings of human culture. Instead, we get ***connected and talk about our technology or offer hardware for sale. Very few true personas have emerged—for fame in the packet community comes instead from authoring a BBS package or owning a big-gun digipeater.

I propose that we make every effort possible to outpace the limitations of our demographics and use these new tools to build an electronic neighborhood.

Every network has to graduate from a period of self-examination to survive. In the early on-line days, there was an obsession with "who we are" and "what all this implies." I remember rhapsodizing into the night on the CompuServe CB simulator, releasing real-time pheromones into the cosmos and tossing thoughts through my electronic window, pausing every few minutes to comment to unseen friends about the magic of this medium. We soon became as gods, dancing through the ether in porpoise-playful alacrity, weaving and

ALWAYS SEEMED TO SHOUT ALL THE TIME. There was an endless buzz of intellectual energy, for we recognized in ourselves a species a-borning.

Then we matured, forming subcultures within subcultures, and life on-line became every bit as real as real estate. When I finally sold the house and moved to a bicycle, there was nary a glitch in my sense of home.

And it wouldn't BE home if it weren't robust, diverse, and infinite in possibilities.

Packet radio grows now on the fringes, enabling another major step in disconnection: the elimination of a wire umbilicus to the closest network node. We have the technology to make a major impact... but do we have the culture? Do we have the diverse range of intellects, the crazies, the impostors, the mad flammers, the playful hackers, the sexy prowlers? Or is packet destined to remain a mix of practical messaging, public service, point-to-multipoint technical exchange, forum for its own documentation, and DX frontier for those burned out on HF contesting?

It's up to all hams. Here's a suggestion: When on packet AND on a big network, take steps to download culture as well as software. Inject network style and commentary onto 145.01; seduce wire-bound networkers into our lair. Proselytize. Build a gateway. Put your SELF onto packet, not just the equipment. Bring women online and flirt with them electronically. Play with software-controlled multi-connects and build multipoint conferencing systems to speed things up. And

then... let the madman inside come out to frolic in the comparative anonymity of life online.

Now... How the System works

First, the introductory overview piece in this series ("On the Road and On the Air," February 1988) is prerequisite reading for the articles that follow. Those who missed the original issue and want a photocopy, send \$1 to the Computing Across America address at the end of this article.

At its heart, of course, the Winnebiko's packet system contains a TNC—the original CMOS unit from Pac-Comm (which is about to be replaced by their nifty new 40mA "Micropower-2" unit). The TNC is standard... but at all levels, its interface to the outside world is non-traditional.

The board is buried deep in the console electronics package, so its status LEDs are remotored to the control panel—along with a switch for power and another to allow a convenient loopback test. Likewise, on-board linear voltage regulators are disabled to save power, since the bike's efficient switching supplies already provide 3, 5, 6, 9, and -12 volts in addition to the twin 12-volt battery buses.

The radio interface is a bit more elaborate than usual, since the Yaesu 290 is a major communications hub in this mobile system. The TNC's push-to-talk line is ORed with both software and handlebar PTT commands before being passed on to the radio; a similar merging occurs with mike and speaker lines through a small "audio nexus board" that includes such things as preamps, a 4-pole filter, mixers, and so on. There's quite enough thought required to run the system without having to manually establish a connection between TNC and radio every time!

Serial Port

Readers may recall from February that all data communication among the bike systems takes place through a "serial crossbar network" under control of the bicycle control processor (BCP). When designing the machine, I knew that I'd want to run packet in the completely different settings, on the road while mobile, and through the HP laptop

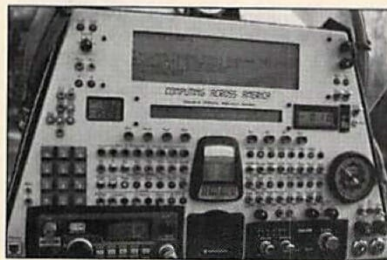


Photo A. The Winnebiko's packet station uses a Pac-Comm Micropower-2 TNC, a Yaesu FT-290 two meter radio, and a Radio Shack Model 100 computer terminal. Its display is an integral part of the bike's console.

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

Ok No Read more (<https://microship.com/privacy-policy/>)

(<https://microship.com/wp-content/uploads/2016/11/73-packeteering-1.jpg>)

I mention all this because hams possess a unique set of tools quite the envy of traditional networkers everywhere. Yet hams have hardly begun to develop the *nth-layer protocols* that govern style and behavior... the essential underpinnings of human culture. Instead, we get ***CONNECTED and talk about our technology or offer hardware for sale. Very few true personas have emerged — for fame in the packet community comes instead from authoring a BBS package or owning a big-gun digipeater.

I propose that we make every effort possible to outpace the limitations of our demographics and use these new tools to build an electronic neighborhood.

Every network has to graduate from a period of self-examination to survive. In the early on-line days, there was an obsession with “who we are” and “what all this implies.” I remember rhapsodizing into the night on the CompuServe CB simulator, releasing real-time pheromones into the cosmos and tossing thoughts through my electronic window, pausing every few minutes to comment to unseen friends about the magic of this medium. We soon became as gods, dancing through the ether in porpoise-playful alacrity, weaving and bobbing among the indelicate compu-boors and DUMB-TERMINAL USERS WHO ALWAYS SEEMED TO SHOUT ALL THE TIME. There was an endless buzz of intellectual energy, for we recognized in ourselves a species a-borning.

Then we matured, forming subcultures within subcultures, and life on-line became every bit as real as real estate. When I finally sold the house and moved to a bicycle, there was nary a glitch in my sense of home.

And it wouldn't BE home if it weren't robust, diverse, and infinite in possibilities.

Packet radio grows now on the fringes, enabling another major step in disconnection: the elimination of a wire umbilicus to the closest network node. We have the technology to make a major impact... but do we have the culture? Do we have the diverse range of intellects, the crazies, the impostors, the mad flammers, the playful hackers, the sexy prowlers? Or is packet destined to remain a mix of practical messaging, public service, point-to-multipoint technical exchange, forum for its own documentation, and DX frontier for those burned out on HF contesting?

It's up to all hams. Here's a suggestion: When on packet AND on a big network, take steps to download culture as well as software. Inject network style and commentary onto 145.01; seduce wire-bound networkers into our lair. Proselytize. Build a gateway. Put your SELF onto packet, not just the equipment. Bring women online and flirt with them electronically. Play with software-controlled multi-connects and build multipoint conferencing systems to speed things up. And then, let the madmen inside come out to frolic in the comparative anonymity of life online. We use the madmen inside come out to frolic in the comparative anonymity of life online, you agree to their use.

Now... How the System works [Read more \(https://microship.com/privacy-policy/\)](https://microship.com/privacy-policy/)

First, the introductory overview piece in this series (*On the Road and On the Air* (<https://microship.com/on-the-road-and-on-the-air-73-amateur-radio/>), February 1988) is prerequisite reading for the articles that follow. At its heart, of course, the Winnebiko's packet system contains a TNC — the original CMOS unit from Pac-Comm (which is about to be replaced by their nifty new 40mA “Micropower-2” unit). The TNC is standard, but at all levels, its interface to the outside world is non-traditional.

The board is buried deep in the console electronics package, so its status LEDs are remoted to the control panel — along with a switch for power and another to allow a convenient loopback test. Likewise, on-board linear voltage regulators are disabled to save power, since the bike's efficient switching supplies already provide 3, 5, 6, 9, and -12 volts in addition to the twin 12-volt battery buses.

The radio interface is a bit more elaborate than usual, since the Yaesu 290 is a major communications hub in this mobile system. The TNC's push-to-talk line is ORed with both software and handlebar PTT commands before being passed on to the radio: a similar merging occurs with mike and speaker lines through a small “audio nexus board” that includes such things as preamps, a 4-pole filter, mixers, and so on. There's quite enough thought required to run the system without having to manually establish a connection between TNC and radio every time!

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

[Read more \(https://microship.com/privacy-policy/\)](https://microship.com/privacy-policy/)

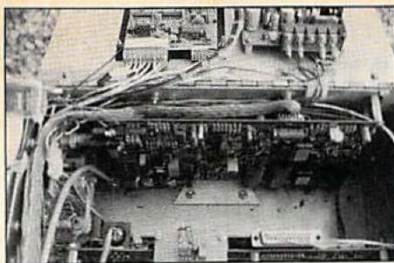


Photo B. The Pac-Comm Micropower-2 TNC, buried deep within the console, actually appears as a node in the serial crossbar network controlled by the bicycle control processor (BCP).

while parked. It was also essential that the hardware architecture allow automation of all packet functions to support the BBS as well as future security and telemetry systems.

The TNC appears as a node in the serial crossbar network, which acts as a central office for all asynchronous data floating around in the bike. This logic can be controlled by 4-bit commands from the BCP or overridden via a front-panel rotary switch. It does its job with an array of 74HC4016 analog switches wired to permit connection between all possible pairs of transmit and receive data lines...with much less overhead than a traditional LAN. From the TNC's standpoint, of course, the only meaningful connections are to the front-panel DB9 (for the HP Portable PLUS) and the Model 100.

In the simplest case, the laptop drives the TNC with no other system components powered up. Things are a little more complex, however, when I'm on the road. Three processors come alive and work together to support a packet QSO.

Besides the TNC, a heavily modified Model 100 with a tree-structured operating system (Traveling Software's Booster Pak) runs BBS and terminal software, displaying its activity on a console LCD. For the last few thousand miles, this machine has been processing under the delusion that it has a normal keyboard, but as I noted back in February, its parallel console port is tied to a chunk of interface logic owned by the BCP.

The Handlebar Keyboard

To understand how this subsystem works, it's necessary to view it in three parts: the handlebar keyboard itself, the interface software, and the hardware that links the BCP to the Model 100.

First, there are a total of eight waterproof keys on the grips, four on each side. The connection between them and the 68HC11 board that runs the bike has become trivial. After riding a few months with a flaky hardware interface, I wised up and tied the push-buttons to an input port through some Schmitt triggers. Why mess around with timing logic and noise-sensitive edge detection when there's a fast and lightly-loaded CPU spinning in a wait loop all day long? The BCP sends a signal back to the handlebars.

The code is straightforward, more or less.

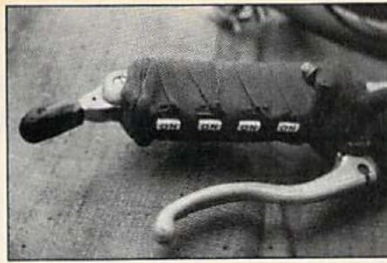


Photo C. The Model 100 accepts input from the eight handlebar keys, which are routed through Schmitt triggers. The BCP watches for activity transitions at the keyboard before further decoding.

As I type in a variant of parallel ASCII, the processor watches for a transition from "no keys down" to "any keys down." As long as that condition exists, it lights a console LED and ORs together any active bits that comprise the character to be latched as soon as "any keys down" returns to "no keys down." This eliminates any tendency toward timing-sensitivity, something that would be a nightmare on bumpy roads with Morse, Braille, or any other one-handed system requiring both setup and strobe actions.

From here, a few checks are made to see if the character is "ignore me" (FF hex) or contains bit combinations that imply a local command to the BCP itself. Then it's passed on to decoding and lookup logic. In short, this maps the handlebar character onto the Model 100's keyswitch matrix, assembling an 8-bit value that contains row address, column address, and a special bit to denote simultaneity (control, shift, graph, and code keys). This byte is then output to hardware with a corresponding flash of another console LED.

But it's not ready for the 100 yet—that machine expects a keyboard full of of passive switches in an X-Y matrix. There's an active 17-pin interface to deal with here, in which decoded column strobes are exhaled by the 100 while row data are inhaled. To emulate all this, a comparator watches the former, matching them against a decoded version of the three column bits in my synthetic character byte. When a match occurs, it causes the appropriately decoded row code to be generated...with another multiplexed into the same cycle if the key has a simultaneous modifier.

Handlebar Key/Character Setup

To keep things reasonably simple for the operator, I shuffled the handlebar bits enough to allow my five strongest fingers, working alone, to generate lower-case alpha. The right little finger causes upper case; the left, control. The left ring finger signifies numeric or special characters (hmmm) and various combinations of those three yield less-common columns on the code chart or a short library of frequently-used words. Alone, the three minor fingers produce return, backspace, and space.

The thumbs handle 2- and 11-meter PTT, airhorns, and sirens. The fleshy outsides of the palms handle two of the three dimensions in the you-agree-to-their-use the entire

hands abort their typing and squeeze like hell when the disc brakes are needed. There are many degrees of freedom remaining, but I need to steer and stay sane as well.

Ergonomics of Mobile Packet

Heh. It's a common question: "How d'ya concentrate on the road when running that computer?" The answer is that the two activities occupy such different parts of my brain that virtually no wetware resource-management is necessary. (I can see it now: my last words captured in a RAMdisk file like the data in a DC-9's "black box." They peel me off a mountain, reconstruct the bike like an airplane to determine the cause of the crash, and find in a file called LSTWRDS.DOC the following text: "Aaaaauuuggghhhhhhhhhhh!")

Actually, running bicycle-mobile packet is pure pleasure—and not just because of the naughty sense of beating trade-offs by inventing new rules and then ignoring them. The pace of packet radio is slow enough that the intrinsic synchrony of a handlebar keyboard (limiting me to about 30 wpm) is not as frustrating as it would be in, say, a GENie real-time conference room. In a perverse way, it fits bicycle touring perfectly....

Imagine chatting digitally with Sourcevoid Dave, himself car-mobile, while pedaling from Palo Alto to Menlo Park. Sending mail to an Internet friend in Sweden while negotiating city streets. Patching together 15,000 miles of NET/ROM links, including two passes through the wormhole, while waiting for KA8ZYW to emerge from a store with an armload of burrito fixin's. Getting directions to W2ICZ's house in Buffalo while pumping along the Lake Erie shore. Watching out a restaurant window as the BBS accumulates mail, the bike's CON, STA, KEY, DCD, PTT, BUSY, and OK lights blinking in the dusk of an unfamiliar town as it opens its electronic doors to a tired wanderer.

Electronic communication is an essential part of today's lifestyle. My perception of ham radio changed a lot when I realized it could be much more than a celebration of technology, a medium for competition, and an off-hours hobby for those not already burned out on knobs and switches. A delicious melding is happening now—a synthesis of widely divergent technologies.

Readers who want to sample network life and have direct access to Steve and his online ramblings (over 40 chapters so far, as well as a dozen or so technical articles), sign onto GENie. Use a modem to call (800) 638-8369, and when connected, hit HHH and wait for the U#= prompt. When it appears, type control-R followed by XJM11878.GENIE and hit RETURN. Have credit-card or checking account data ready—the system goes through a one-time sign-up and then bills the user \$5/hour for subsequent non-business-hours access. Steve can be reached by electronic mail as WORDY; Larry Ledlow, the editor of 73, can be reached online as L.LEDLAW or EDITOR73. Finally, please note that my call has changed from KA8OVA to N4RVE. 73

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

Serial Port

Readers may recall from February that all data communication among the bike systems takes place through a “serial crossbar network” under control of the bicycle control processor (BCP). When designing the machine, I knew that I’d want to run packet in two completely different settings: on the road while mobile, and through the HP laptop while parked. It was also essential that the hardware architecture allow automation of all packet functions to support the BBS as well as future security and telemetry systems.

The TNC appears as a node in the serial crossbar network, which acts as a central office for all asynchronous data floating around in the bike. This logic can be controlled by 4-bit commands from the BCP or overridden via a front-panel rotary switch. It does its job with an array of 74HC4016 analog switches wired to permit connection between all possible pairs of transmit and receive data lines...with much less overhead than a traditional LAN. From the TNC’s standpoint, of course, the only meaningful connections are to the front-panel DB9 (for the HP Portable PLUS) and the Model 100.

In the simplest case, the laptop drives the TNC with no other system components powered up. Things are a little more complex, however, when I’m on the road. Three processors come alive and work together to support a packet QSO.

Besides the TNC, a heavily modified Model 100 with a tree-structured operating system (Traveling Software’s *Booster Pak*) runs BBS and terminal software, displaying its activity on a console LCD. For the last few thousand miles, this machine has been processing under the delusion that it has a normal keyboard, but as I noted back in February, its parallel console port is tied to a chunk of interface logic owned by the BCP.

The Handlebar Keyboard

To understand how this subsystem works, it’s necessary to view it in three parts: the handlebar keyboard itself, the interface software, and the the hardware that links the BCP to the Model 100.

First, there are a total of eight waterproof keys on the grips, four on each side. The connection between them and the 68HC11 board that runs the bike has become trivial. After riding a few months with a flaky hardware interface, I wised up and tied the push buttons to an input port through some Schmitt triggers. Why mess around with timing logic and noise-sensitive edge detection when there’s a fast and lightly-loaded CPU spinning in a wait loop all day long? The BCP now polls an input port to see what’s happening back on the handlebars.

The code is straightforward, more or less. As I type in a variant of parallel ASCII, the processor watches for a transition from “no keys down” to “any keys down.” As long as that condition exists, it lights a console LED and ORs together any active bits that comprise the character to be latched as soon as “any keys down” returns to “no keys down.” This eliminates any tendency

toward timing-sensitivity, something that would be a nightmare on bumpy roads with Morse, Braille, or any other one-handed system requiring both setup and strobe actions.

From here, a few checks are made to see if the character is “ignore me” (FF hex) or contains bit combinations that imply a local command to the BCP itself. Then it’s passed on to decoding and lookup logic. In short, this maps the handlebar character onto the Model 100’s keyswitch matrix, assembling an 8-bit value that contains row address, column address, and a special bit to denote simultaneity (control, shift, graph, and code keys). This byte is then output to hardware with a corresponding flash of another console LED.

But it’s not ready for the 100 yet — that machine expects a keyboard full of passive switches in an X-Y matrix. There’s an active 17-pin interface to deal with here, in which decoded column strobes are exhaled by the 100 while row data are inhaled. To emulate all this, a comparator watches the former, matching them against a decoded version of the three column bits in my synthetic character byte. When a match occurs, it causes the appropriately decoded row code to be generated... with another multiplexed into the same cycle if the key has a simultaneous modifier.

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

[Read more \(https://microship.com/privacy-policy/\)](https://microship.com/privacy-policy/)

Winnebiko Handlebar Keyboard Coding

last revised -- 10/5/86

ZON	SET	4	2	1	Alone	SHF	CTR	NUM	CTR-NUM	CTR-S
0	0	0	0	0		RET	BS	SPC		(HC11)
0	0	0	0	1	a	A	^A	!	f1	
0	0	0	1	0	b	B	^B	"	f2	
0	0	0	1	1	c	C	^C	#	f3	
0	0	1	0	0	d	D	^D	\$	f4	
0	0	1	0	1	e	E	^E	%	f5	
0	0	1	1	0	f	F	^F	&	f6	
0	0	1	1	1	g	G	^G	'	f7	
0	1	0	0	0	h	H	^H	(f8	
0	1	0	0	1	i	I	^I)	PASTE	
0	1	0	1	0	j	J	^J	*	LABEL	
0	1	0	1	1	k	K	^K	+		
0	1	1	0	0	l	L	^L	,		
0	1	1	0	1	m	M	^M	-	TAB	
0	1	1	1	0	n	N	^N	.		
0	1	1	1	1	o	O	^O	/	ESC	
1	0	0	0	0	p	P	^P	0	<	
1	0	0	0	1	q	Q	^Q	1	>	
1	0	0	1	0	r	R	^R	2	^	
1	0	0	1	1	s	S	^S	3	v	
1	0	1	0	0	t	T	^T	4	shf-<	
1	0	1	0	1	u	U	^U	5	shf->	
1	0	1	1	0	v	V	^V	6	shf-^	
1	0	1	1	1	w	W	^W	7	shf-v	
1	1	0	0	0	x	X	^X	8	ctr-<	
1	1	0	0	1	y	Y	^Y	9	ctr->	
1	1	0	1	0	z	Z	^Z	:	ctr-^	
1	1	0	1	1		[ESC	;	ctr-v	
1	1	1	0	0		@		<		
1	1	1	0	1]		=		
1	1	1	1	0		^		>		
1	1	1	1	1	DEL			?		

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

All keys depressed - Read <https://microship.com/privacy-policy/>

Handlebar Key/Character Setup

To keep things reasonably simple for the operator. I shuffled the handlebar bits enough to allow my five strongest fingers, working alone, to generate lower-case alpha. The right little finger causes upper case: the left, control. The left ring finger signifies numeric or special characters (hmmm) and various combinations of those three yield less-common columns on the code chart or a short library of frequently-used words. Alone, the three minor fingers produce return, back space, and space.

The thumbs handle 2- and 11-meter PTT, air horns, and sirens. The fleshy outsides of the palms handle two of the three dimensions in the 54-speed transmission, and the entire hands abort their typing and squeeze like hell when the disc brakes are needed. There are many degrees of freedom remaining, but I need to steer and stay sane as well.

Ergonomics of Mobile Packet

Heh. It's a common question: "Howd'ya concentrate on the road when running that computer?" The answer is that the two activities occupy such different parts of my brain that virtually no wetware resource-management is necessary. (I can see it now: my last words captured in a RAMdisk file like the data in a DC-9's "black box." They peel me off a mountain, reconstruct the bike like an airplane to determine the cause of the crash, and find in a file called LSTWRDS.DOC the following text: "Aaaaauuuuggghhhhhhhh!")

Actually, running bicycle-mobile packet is pure pleasure — and not just because of the naughty sense of beating trade-offs by inventing new rules and then ignoring them. The pace of packet radio is slow enough that the intrinsic synchrony of a handlebar keyboard (limiting me to about 30 wpm) is not as frustrating as it would be in, say, a GENie real-time conference room. In a perverse way, it fits bicycle touring perfectly....

Imagine chatting digitally with Sourcevoid Dave (<https://microship.com/roughing-palo-alto/>), himself car-mobile, while pedaling from Palo Alto to Menlo Park. Sending mail to an Internet friend in Sweden while negotiating city streets. Patching together 15,000 miles of NET/ROM links, including two passes through the wormhole, while waiting for KA8ZYW to emerge from a store with an armload of burrito fixin's. Getting directions to W2ICZ's house in Buffalo while pumping along the Lake Erie shore. Watching out a restaurant window as the BBS accumulates mail, the bike's CON, STA, KEY, DCD, PTT, BUSY, and OK lights blinking in the dusk of an unfamiliar town as it opens its electronic doors to a tired wanderer.

Electronic communication is an essential part of today's lifestyle. My perception of ham radio changed a lot when I realized it could be much more than a celebration of technology, a medium for competition, and an off-hours hobby for those not already burned out on knobs and switches. We use packets to ensure that we give hobby for those not already burned out on knobs and switches. you agree to their use.

A delicious melding is happening now — a synthesis of widely divergent technologies.

Ok No Read more (<https://microship.com/privacy-policy/>)

APRIL 1988
ISSUE #331

73 AMATEUR RADIO

International Edition

USA \$2.95
CAN. \$3.95
A WGE Publication

Ham Radio Responds!

Exciting features and
essential projects —
Be Prepared!

Exclusive Profile

Leonid Labutin UA3CR

Reviews

- Kenwood's TS-140 sets a new standard
- ICOM's IC-960 —Futuristic FM
- Amp Supply TR-110 —tops on ten



We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

Ok No Read more (<https://microship.com/privacy-policy/>)

(<https://microship.com/wp-content/uploads/2016/11/73-april-1988-cover.jpg>)

Related

(<https://microship.com/life-on-a-megacycle/>)

Life on a Megacycle

(<https://microship.com/life-on-a-megacycle/>)

November 1, 1988

In "Articles by SKR"



(<https://microship.com/packet-radio-high-tech-nomadics/>)

Packet Radio and High-Tech Nomadics (<https://microship.com/packet-radio-high-tech-nomadics/>)

October 1, 1989

In "Articles by SKR"



(<https://microship.com/behemoth-visits-radio-fun-73/>)

BEHEMOTH Visits Radio Fun/73 (<https://microship.com/behemoth-visits-radio-fun-73/>)

August 1, 1992

In "Media - BEHEMOTH"

Posted in [Articles by SKR](https://microship.com/category/magazine-articles-by-skr/) (<https://microship.com/category/magazine-articles-by-skr/>) and tagged [73 Magazine](https://microship.com/tag/73-magazine/) (<https://microship.com/tag/73-magazine/>)

← [High Trek – Charlotte](https://microship.com/high-trek-charlotte/) (<https://microship.com/high-trek-charlotte/>)

[Pedaling takes back seat with high-tech cycle – Greensboro](https://microship.com/pedaling-takes-back-seat-greensboro/) → (<https://microship.com/pedaling-takes-back-seat-greensboro/>)

Affiliate links

Nomadic Research Labs is a participant in the Amazon Services LLC Associates Program, an affiliate advertising program designed to provide a means for sites to earn advertising fees by advertising and linking to Amazon.com.

[Privacy](#)

[Cookie](#)

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

Ok

No

Copyright 1983-2020 Nomadic Research Labs

[Read more \(https://microship.com/privacy-policy/\)](https://microship.com/privacy-policy/)

[Contact Steve \(https://microship.com/contact/\)](https://microship.com/contact/)

[Privacy Policy \(https://microship.com/privacy-policy/\)](https://microship.com/privacy-policy/)

[Terms and Conditions \(https://microship.com/terms/\)](https://microship.com/terms/)

[Refund and Return Policy \(https://microship.com/refund/\)](https://microship.com/refund/)

☺

We use cookies to ensure that we give you the best experience on our website. By continuing to use this website, you agree to their use.

[Ok](#) [No](#) [Read more \(https://microship.com/privacy-policy/\)](https://microship.com/privacy-policy/)