Celebrating the 99th issue of TI*MES

Winter 2007 / 2008

TI*MES index.

Committee Members

Chairman Trevor Stevens. 249 Southwell Road East, Rainworth, NG21 0BN Phone: 01623 406133 chairman@ti99ug.co.uk	**Media Librarian** Francesco Lama 48 Mayfair Road Cowley, Oxford. OX4 3SR disklibrary@ti99ug.co.uk
General Secretary Richard Twyning 64 Hornbeam Way, Kirkby-In-Ashfield, NG17 8RL Phone: 07767 44 56 58 treasurer@ti99ug.co.uk	**International Roving Reporter** Berry Harmsen berry.harmsen@wxs.nl

Supporting the **TI-99/4** and **/4A**, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.

Disclaimer

The views expressed in the articles in this magazine are those of the individual author, and not necessarily the view of the magazine or the group.

Email membership terms and conditions! J

At this point we would like to give a warning to those who have subscribed with email membership. You have expressed this preference because you want your TI*MES magazine to be delivered by a more reliable medium than the Royal Mail! However, as you must realise, PC file formats these days are bloated beyond belief and the Microsoft Word file of a newsletter could be as big as 18 Megabytes!!!
PDF files do compress the file size down, but they may still be a considerable size!

Receiving large attachments these days is no big deal, as Yahoo.co.uk offer a free email service which gives you UNLIMITED storage!!!!

If you specify an email address for your email membership then you **_MUST MAKE SURE_** that the email account has **_sufficient capacity_** to be able to receive these attachments!!!

It's not all doom and gloom though! Trevor and I have been pondering converting the magazine to HTML format and putting it on the web site so that people could read it online. Trevor already has the web code to allow us to make it password protected from non-signed-up visitors! We also need to make sure that it only uses bog-standard HTML and GIF or JPEG images so that it is available to a wider audience. Trevor has the habit of automatically using loads of flash and applets on his websites, but I think we need to make things a bit simpler so it can be accessed by a wider range of web browsers. **J**

Important Information

Membership Renewals

If you have access to the interweb you are able to pay your group membership directly from the user group website using your PayPal account.
For those who aren't in the know, PayPal is a web site that allows you to send and receive money internationally. It's excellent, and has recently been purchased by a little company called **ebay!**

The group's web address is **www.ti99ug.co.uk** then just click the "**join TI user group**" button!
While still on the subject of the website, don't forget the TI picture book where you can see photos from TI events. Check back regularly for updates.
The website now has a passworded zone where you can read or download previous and current issues of TI*MES magazine.

Supporting the **TI-99/4** and **/4A**, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.

The skype Page

Welcome to a new regular page in the magazine. Trevor and I did have our Skype contact details on the committee members page, however I've decided to add a separate page to try and promote Skype since it's free and will enable us to keep in touch with TI'ers around the world much more easily!
Hopefully having this dedicated page will raise the profile of Skype and encourage more members to use it.
If we have enough we could have big online conference which is what we've talked about doing in the past.

Name	Skype Username
Richard Twyning	richardtwyning
	richardtwyningmobile
Trevor Stevens	trevorstevensmegatech
Mark Wills	markrobertwills
Berry Harmsen	berry.harmsen
Tom wills	twills44

I'm deliberately keeping this page sparse as it's your job to fill it by contacting us on Skype and letting us have your username! Also, read the small print below!!!
I don't think Skype will work on Windows 98, but it's available for most other things including Pocket PC's running Windows Mobile.   If you've got an older PC then you might want to consider installing Linux, because there's a version of SKYPE available for it!

Please pay careful attention to the usernames as some of them may contain full stops between forename and surname and they won't work unless they are entered correctly!!!

From the Chairman's Chair

By Trevor Stevens

I would like to start by saying to all TIer's every where, a Merry Christmas and a happy New Year. The TI-99/4a enters into another year with perhaps the best chance of seeing another dawn.

I have been looking about and there are projects still being done for the TI which really heartens me to see that there is still dedicated members of the TI community that wish not to see the TI-99/4A.

So 2008 should be a nice year to see a few more of those project completed and developed. So watch our mag and you will see all the latest news as we get it.

Just to give you a taster of an older project here is one to speed up the elephants trunk…

This is a pretty easy project and though it won't make a night or day difference, any little speed gain helps. This "Tune up" was explained to me by Guru Richard Bell. As you know this board is what attaches the console to the PEB, and with projects like the crystal upgrade or 32K on the 16 bit bus which speed up the console, getting that data to and from the PEB just a little bit quicker can't be a bad thing!

Keep in mind that is mod is more about "bottle neck relief" than speed increase for a stock console. In other words this mod serves to allow higher speed in conjunction with the 16 bit memory speed and over clocked crystal speeds.
This is one of those projects where you may not see much, if any, advantage, but for about £1.50 in parts, and about 30 minutes soldering it really can't hurt!

What we are doing here is replacing the three 74LS244 buffers with three 74HCT244 buffers. Make sure that the new 244's are marked HCT.
These are "HIGH SPEED" buffers and the T makes them compatible with TTL inputs.

While you are at it for a few pence more go ahead and replace the two electrolytic capacitors as with the age they are at they probably are drying out.

I have all the parts in question marked by yellow boxes in the pictures below.

So what you will need are:

   3 * 74HCT244 IC's
   4 * 22 µF electrolytic capacitors 25v-35v (35v preferred)

⚠️ **Please make sure that all wiring and soldering is double checked before re-applying power.**
     Though computer components are pretty hardy some do not take to being wired up backwards or incorrectly.

I have done my best to make sure that all diagrams and instructions given here are correct, but I can not be responsible for any damage an incorrect upgrade might cause.

De solder the parts shown, and take care not to remove any of the foil. Replace with the new parts, paying attention to the orientation of the chips, and the polarity of the capacitors, see **figure 1**.
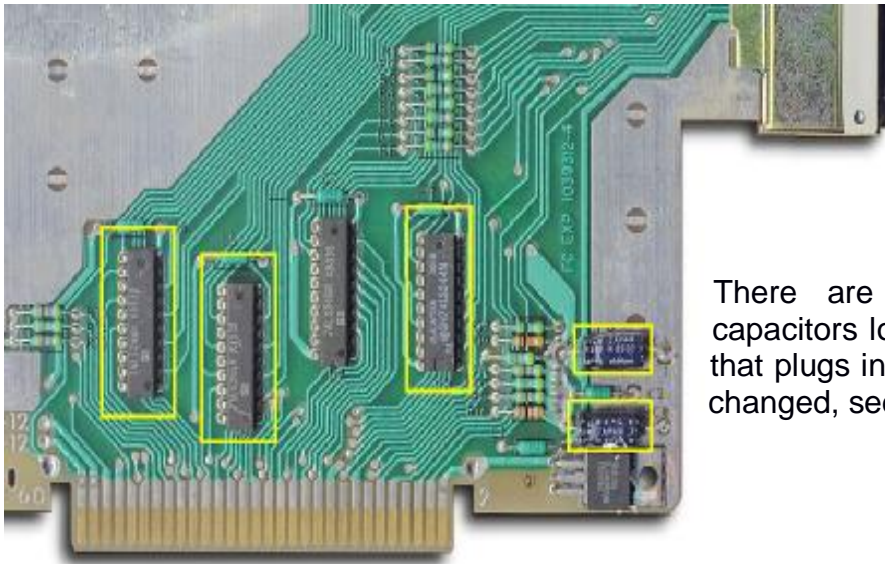
Z Z Z Z

There are also two of the same type capacitors located in the end of the flex cable that plugs into the console that should also be changed, see figure 2.

Figure 2.

` ` ` `

The project is brought to you by kind permission of Ron Reuter of www.mainbyte.com

To put you in the mood to drag out your TI we have included in this issue a bumper Christmas present of listings for you to type in. The range is from the small music typo through to assembly stuff; it should keep all you TI tappers happy.

You will have seen in the last issue the number of deaths we are getting in our community. However we have been getting some new members which is really nice to see, some coming from overseas.

I hope you have enjoyed this year's magazines and Group activities as much as I have. I know that I have been involved in the printing etc but there others which do a lot of work.

So on behalf of the Group I would like to thank all active members for their dedication and input for keeping the TI*MES and TIUG (UK) alive.

Merry Christmas to you all!
& happy New Year

Fctn Quit.

Supporting the **TI-99/4** and **/4A**, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.

From Kirkby With Love –
By Richard Twyning

Dear TI'er,

We hope you liked the last issue despite the fact that it conveyed some bad news.

Hopefully we can cheer you up now as promised by giving you lots of bits and bobs to type in over Christmas!

OK, so what's the big idea with the title of my article? For the keen eyed amongst you who read everything like a hawk, you will have spotted that my address has changed! For the less observant, I think we need some small print J

Small print…

Richard Twyning's address has changed….

64 Hornbeam Way, Kirkby-In-Ashfield, Nottinghamshire. NG17 8RL

Trust me, there still seem to be things that people miss in the magazine, so hopefully they can't miss that!

Well, in this issue I did promise lots of programs to type in, so without messing about, let me have a look at what I've found...

The first program was found by our own faithful Mark Wills J

Pocket sunrise

```
Subject:   [TI-99/4A] Nice little XB Program ;-)
From:      mark wills <markrobertwills@yahoo.co.uk>
Date:      Wed, 14 Nov 2007 08:29:59 -0800 (PST)
To:        ti99-4a@yahoogroups.com
```

Very nice, imaginative use of CHAR and scrolling... Paste into Classic99 for a smile ;-)

Mark.

```
100 REM *****************         , 2*(8-EIGHTH))
110 REM * POCKET SUNRISE *        450 SKY=(CHSET+3)*8+EIGHTH-1
120 REM *****************         460 CALL CHAR(SKY, PATTERN$)
130 REM BY S. T. HOLL             470 CALL HCHAR(23, 1, SKY, 32)
140 REM 99'ER VERSION 2.11.1XB    480 IF OFF+(1-ONN)THEN 560
150 DATA 32, 42, 2, 3, 88, 6, 0, 0, 0   490 SUNHI=SUNHI+1
, 10                              500 SWIDE=SUNHI
160 READ SKY, STAR, FGC, CHSET    510 IF SWIDE<5 THEN 530
, SUN, SUNRISE, SUNHI, ONN, OFF, WC  520 SWIDE=9-SWIDE
170 CALL CLEAR                    530 CALL HCHAR(23
180 CALL SCREEN(2)                , 17-SWIDE, SUN, 2*SWIDE)
190 GOSUB 630                     540 IF SUNHI<8 THEN 560
200 CALL COLOR(1, 2, 2)           550 OFF=1
210 CALL COLOR(2, 16, 2)          560 PRINT :
220 CALL COLOR(8, 11, 11)         570 GOSUB 600
230 FG$="FFFFFFFFFFFFFFFF"        580 NEXT EIGHTH
240 BG$="0000000000000000"        590 GO TO 320
250 FOR DARK=1 TO 50              600 FOR TICK=1 TO 10
260 CALL HCHAR(23, 1, SKY, 32)    610 NEXT TICK
270 CALL HCHAR(23, RND*30+1       620 RETURN
, STAR, 1)                        630 CALL MAGNIFY(4)
280 PRINT :                       640 RANDOMIZE
290 GOSUB 860                     650 CALL CHAR(140
300 GOSUB 600                     , "FFFFDF1E1E1FFC1")
310 NEXT DARK                     660 CALL CHAR(141
320 BGC=FGC                       , "C1C1C1C1FFFFFFFF")
330 DATA 15, 9, 11, 4, 5, 6, 6, 6, 6, 0   670 CALL CHAR(142
340 READ FGC                      , "FFFBF8F8787FF83")
350 IF FGC<>0 THEN 370            680 CALL CHAR(143
360 GO TO 360                     , "83838383FFFFFFFF")
370 CHSET=CHSET+1                 690 FOR I=1 TO 5
380 IF CHSET<7 THEN 400           700 SKYLINE(I)=185
390 CHSET=2                       710 NEXT I
400 CALL COLOR(CHSET, FGC, BGC)   720 FOR SP=1 TO 10
410 IF FGC<>SUNRISE THEN 430      730 I=INT(1+RND*5)
420 ONN=1                         740 SKYLINE(I)=SKYLINE(I)-32
430 FOR EIGHTH=1 TO 8             750 CALL SPRITE(#SP, 140, 2
440 PATTERN$=SEG$(FG$, 1          , SKYLINE(I), 32+32*I)
, 2*EIGHTH)&SEG$(BG$, 1           760 NEXT SP
```

```
770 CALL CHAR(132
,"0000000000000080")
780 CALL CHAR(133
,"6308080001000000")
790 CALL CHAR(134
,"0000000000000080")
800 CALL CHAR(135
,"000000001C61010")
810 CALL CHAR(136
,"0000000000000000")
820 CALL CHAR(137
,"0014000000000000")
830 CALL CHAR(138
```

```
,"0000000000000000")
840 CALL CHAR(139
,"0000000000002800")
850 RETURN
860 IF DARK-12*INT(DARK/12)<>2
THEN RETURN
870 WC=WC+1 :: UP=-2-RND*8 ::
RT=-15+RND*30
880 CALL SPRITE(#WC,132,2,161
,83,UP,RT)
890 CALL SPRITE(#WC+7,136,7
,161,83+RT/4,UP,RT)
900 RETURN
```

BluOnyx Update

Before the next item I've got a little update on the BluOnyx
that I introduced at this year's Treff and in my last article.

There is a slight setback in this project as the BluOnyx itself
has undergone a setback. LSI had planned to market the device
itself, but they had a change of heart.
They were obviously locked in some intensive negotiations with third parties, as Fadi Al-
Rafaee couldn't talk to me from August onwards and said he would contact me in
November.
Then he contacted me to say that rather than market the device themselves they had
decided to sell it and all of their research to a third party.

This isn't so serious as it might first sound. I know the reaction that BluOnyx has created
on the internet and its new owner certainly can't afford to sit on it indefinitely.
I'm sure we'll see something of it very soon.

Seagate Dave

If things do go all pear shaped with the BluOnyx then there's
still no need to worry as we'll use its rival device!

Yes, there's more than one company in the Mobile
Content Server market!

You might have heard of this company. SEAGATE.
Yes, the world's leading manufacturer of hard disk drives has
also produced one of these devices. As yet I am unsure of the operating
system that this device uses, but it does in fact have more hard disk capacity!
The only disadvantage of this device, if you can call it that is that it doesn't have an SD

card slot. However, it does have a massive 60 Gigabyte capacity!

Description

Powerful Bluetooth 2.0 and Wi-Fi connection
· Slip into a pocket, purse or backpack—stays connected up to 30 feet (9.1 meters) away

Massive 30-GB or 60-GB capacity
· 60 GB holds 15,000 songs1, 60 movies2 or 30,000 photos

High power efficiency
· Delivers hours of media streaming performance

Uses Seagate® hard drives with free-fall protection
· Withstands the accidents and mishaps of everyday mobile use

Wide connectivity options
· Connects to a variety of content sources and compatible with most mobile phone operating systems
· Uses three standard connectivity protocols: USB 2.0, Bluetooth 2.0 EDR and 802.11 b/g

Open development platform – we will still be able to develop TI-99/4A solutions as with the BluOnyx
· Gives developers the opportunity to develop new games, applications and content utilizing increased storage
· Provides for the development of applications that utilize any device with Bluetooth, Wi-Fi or USB connectivity

The Seagate Advantage
Consumers demand more storage in all aspects of their lives—especially for mobile entertainment and information.
Many consumers carry intelligent, connectable devices with them anywhere they go, want access to broadband networks wherever they are, and use technology to obtain their entertainment wherever they are. But creating the right solution for the consumer is difficult. Syncing mobile devices to hub devices in the home is a challenge due to multiple software and connectivity challenges. As content is obtained from many different sources, it becomes decentralized—more difficult for consumers to have the content they want where they want it, on the right device. And managing mobile content is a labour-intensive manual process with the need to manage limited storage on flash-based devices. It is also difficult to increase the amount of embedded system storage to meet scaling consumer demand. What's needed is a way toward unified content management—a state where the consumer has access to content wherever and whenever they want, on any device. Seagate D.A.V.E. technology provides a solution for unified content management. It centralizes the consumer's primary collection of digital content and allows standards-based connectivity to a wide variety of devices. Storage is separated from individual devices, and the consumer can upgrade storage as the Seagate D.A.V.E. platform scales in capacity. This solution is customizable and designed to be branded as a technology accessory by mobile handset OEMs, automobile and automotive component manufacturers, media centre and DVR brands, and other companies looking to add a mobile storage accessory to their consumer solutions. Seagate Design Service Centres will assist in compatibility testing and certification.

For mobile device designers and manufacturers, the Seagate D.A.V.E. platform means lower BOM costs and new revenue opportunities. For home device designers and manufacturers, it brings new revenue models, enhanced media centre content management and life-cycle separation. And for consumers, the promise is simplified content management and the ability to make all their content mobile. The Seagate D.A.V.E. platform gives everybody what they want.

Supporting the **TI-99/4** and /4A, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.

Technical Details

· **Dimensions 2.6 x 3.7 x 0.71 in/65 x 95 x 18 mm**
· **Weight 3.88 oz/110 g**
· **Connectivity Bluetooth, WiFi**
· **Platform "Open" Source Platform**
· **Battery Type 3.7V - Capacity ~940mAh - Charging Cycles (min) 500**
· **Nonoperating Temperature (less than 1 month) –20°C to 45°C**
· **Nonoperating Temperature (less than 6 months) –20°C to 35°C**
· **Nonoperating Humidity (noncondensing) 5% to 95%**
· **Operating Humidity 5% to 90%**
· **Nonoperating Altitude –60 m to 12,000 m**
· **Operating Altitude –60 m to 3000 m**
· **Operating Temperature (discharging battery) 0°C to 60°C**
· **Operating Temperature (charging battery) 0°C to 45°C**

Another remarkable thing about Seagate DAVE is that you can already pre-order one from expansys.com!

Disassembler in BASIC

```
From:        Stuart Conner <ti99@...>
To:          ti99-4a@yahoogroups.com
Sent:        Wednesday, 8 August, 2007 12:55:18 AM
Subject:     [TI-99/4A] Re: Disassembler in BASIC?
```

I wrote a disassembler in XB as well. My approach was slightly different to Thiery's suggestion, in that I had data statements specifying the 'fixed' part of each op-code in binary, the instruction format, and the length of the binary string (quicker to read it than calculate it each time ...) and read this into an array:

```
320 DATA "LI  "," 000000100000"  ,8,12        380 DATA "STWP"," 0000001 0101 0",8,12
330 DATA "AI  "," 000000100010"  ,8,12        390 DATA "LWPI"," 0000001 0111 00000"
340 DATA "ANDI"," 0000001001 0 0",8,12            ,8,16
350 DATA "ORI "," 0000001001 10" ,8,12        400 DATA "LIMI"," 00000011000 00000"
360 DATA "CI  "," 0000001 01000" ,8,12            ,8,16
370 DATA "STST"," 0000001 0110 0",8,12
```

etc.

To decode an instruction, I converted the instruction's hex value into a binary string then used SEG$ to find the matching op-code. I found it easier to work using a binary string as I can then use SEG$ to isolate the various bit fields within the string (without having to do logical ANDs then dividing the result and so on).

Stuart.

```
--- In ti99-4a@yahoogroups.com, mark wills <markrobertwills@...> wrote:
> >
> > Thanks Stuart, there's some really neat code in here.
> >
```

```
> > Would you have any objection if this program was published
> > in TI*MES (the UK users group magazine)?
> > You don't have the code for the 40 column stuff do you?
> > Or is it embedded in the program (off to have a look).
> >
> > Mark
> >
> > Mark Wills, MSc MIAP
> > http://www.markwills.co.uk
> >
> > ----- Original Message ----
> > From: Stuart Conner <ti99@...>
> > To: ti99-4a@yahoogroups.com
> > Sent: Wednesday, 8 August, 2007 6:54:48 PM
> > Subject: [TI-99/4A] Re: Disassembler in BASIC?
> >
> >
```

Subject: [TI-99/4A] Re: Disassembler in BASIC?
From: "Stuart Conner" <ti99@stuartconner.me.uk>
Date: Wed, 08 Aug 2007 22:35:32 -0000
To: ti99-4a@yahoogroups.com

No problems putting it in TI*MES. One always gets a nice
warm feeling seeing one's name in print!
Code for 40 column stuff copied below. It needs to be
compiled into the file DISASS/O.

```
=========
*
*CODE COMPATIBLE WITH XB WITH 32K RAM ONLY.
*LOAD USING CALL LOAD("DSK3.DISASS/O").
*

*
*DEFINE ENTRY POINTS.
*
      DEF TEXT
      DEF GRPHIC
      DEF PRINT

*
*XB REQUIRES EQU's INSTEAD OF REF's.
*SEE E/A MANUAL SECTION 24.4.8.
*
STRREF EQU >2014         GET STRING PARAMETER.
VSBW   EQU >2020         VDP SINGLE BYTE WRITE.
VMBW   EQU >2024         VDP MULTIPLE BYTE WRITE.
VMBR   EQU >202C         VDP MULTIPLE BYTE READ.
VWTR   EQU >2030         VDP WRITE TO REGISTER.

STATUS EQU >837C

*
*WORKSPACE - SEE E/A MANUAL SECTION 24.11.2.
*

*
*BASE ADDRESS IN LOW MEMORY, AFTER UTILITIES.
*
      AORG >2800
```

```
*-------------------------------------------------
*SWITCH TO TEXT MODE (40 X 24) AND CLEAR SCREEN.
*CALL LINK("TEXT")
*-------------------------------------------------


*SET VDP REGISTER 1 TO SET GRAPHIC MODE BUT WITH
*DISPLAY DISABLED UNTIL WE HAVE SET UP THE SCREEN.

TEXT  LI R0,>01B0      VDP REGISTER 1 - TEXT MODE WITH DISPLAY DISABLED.
      BLWP @VWTR        VDP WRITE TO REGISTER.

*SET VDP REGISTER 2 TO RELOCATE SCREEN IMAGE TABLE.

      LI R0,>020E       RELOCATE SCREEN IMAGE TABLE TO >3800 - >3BBF.
      BLWP @VWTR        VDP WRITE TO REGISTER.

*CLEAR SCREEN.

      LI R0,>3800       START WRITING AT START OF SCREEN IMAGE TABLE.
      LI R1,>8000       CHAR CODE FOR '<SPACE>' - >2000 + XB OFFSET OF >6000.
      LI R2,960-1       960 CHARS TO DO - BUT DO 1 OUTSIDE LOOP.
      BLWP @VSBW        WRITE ONE BYTE.
CLR1  MOVB R1,@>8C00    WRITE SUCCESSIVE BYTES WITHOUT DOING A BLWP - QUICKER.
      DEC R2
      JNE CLR1          CONTINUE UNTIL WRITTEN ALL CHARS.

*SET VDP REGISTER 7 TO SET SCREEN COLOURS.

      LI R0,>07F5       VDP REGISTER 7 - COLOUR: WHITE ON LIGHT BLUE.
      BLWP @VWTR        VDP WRITE TO REGISTER.

*SET VDP REGISTER 1 TO ENABLE DISPLAY.

      LI R0,>01F0       VDP REGISTER 1 - TEXT MODE WITH DISPLAY ENABLED.
      BLWP @VWTR        VDP WRITE TO REGISTER.
      SWPB R0           WRITE REGISTER 1 CONTENTS TO >83D4.
      MOVB R0,@>83D4    SEE E/A MANUAL PAGE 326.

*SET PRINT POSITION TO 1ST CHAR.

      LI R0,>3800       START PRINTING AT START OF SCREEN IMAGE TABLE.
      MOV R0,@CURRLN    SAVE PRINT POSITION BETWEEN SUCCESSIVE PRINT CALLS.

*RETURN TO XB.

      CLR R0            CLEAR STATUS.
      MOVB R0,@STATUS   SEE E/A MANUAL SECTION 24.11.2.
      B *R11

*---------------------------------
*SWITCH TO GRAPHICS MODE (32 X 24).
*CALL LINK("GRPHIC")
*---------------------------------

*SET VDP REGISTER 1 TO SET GRAPHIC MODE.

GRPHIC LI R0,>01E0      VDP REGISTER 1 - GRAPHICS MODE.
      BLWP @VWTR        VDP WRITE TO REGISTER.
      SWPB R0           WRITE REGISTER 1 CONTENTS TO >83D4.
      MOVB R0,@>83D4    SEE E/A MANUAL PAGE 326.

*SET VDP REGISTER 2 TO RESET SCREEN IMAGE TABLE LOCATION.

      LI R0,>0200       RELOCATE SCREEN IMAGE TABLE TO >0000.
      BLWP @VWTR        VDP WRITE TO REGISTER.

*RETURN TO XB.

      CLR R0            CLEAR STATUS.
      MOVB R0,@STATUS   SEE E/A MANUAL SECTION 24.11.2.
      B *R11
```

```
*------------------------------------------
*PRINT LINE OF TEXT, SCROLLING IF NEC ESSARY.
*CALL LINK("PRINT",TXTSTRING)
*------------------------------------------

*DEFINE TEXT BUFFER.

TXTBUF BSS 41               40 CHARACTER (1 LINE) TEXT BUFFER + 1 BYTE FOR LENGTH.
H60    BYTE >60             XB CHAR OFFSET.
H80    BYTE >80             SP ACE CHAR.
       EVEN
CURRLN BSS 2                STORES NEXT CHARACTER POSITION TO PRINT TO.
*                          USE CHAR COUNT INSTEAD OF LINE COUNT SO WE DON`T NEED
*                          TO MULTIPLY LINE NUMBERS BY LINE LENGTHS.


*CHECK IF NEED TO SCROLL, THEN SCROLL IF NECESSARY
*DON'T SCROLL TOP TWO LINES AS THESE ARE COLUMN HEADERS.

PRINT  MOV @CURRLN,R0       GET NEXT PRINT POSITION.
       CI R0,>3BBF          CHECK IF OFF END OF SCREEN.
       JLE FILL             NO NEED TO SCROLL - SO JUMP.

       LI R0,>3800+120      START COPYING FROM 4TH LINE (MOVE IT UP TO
3RD LINE).
       LI R1,TXTBUF         START ADDRESS OF TEXT BUFFER.
       LI R2,40             DO 1 LINE (40 CHARS) AT A TIME.
LOOP5  BLWP @VMBR           COPY LINE TO BUFFER.
       AI R0,-40            MOVE TO LINE ABOVE.
       BLWP @VMBW           WRITE FROM BUFFER BACK TO LINE.
       AI R0,80             MOVE DOWN TO NEXT LINE TO COPY.
       CI R0,>3BBF          DONE ALL LINES?
       JLE LOOP5            NO - LOOP.

       LI R0,>3BC0-40       SET NEXT PRINT POSITION TO LAST LINE.
       MOV R0,@CURRLN       SAVE PRINT POSITION.


*FILL TEXT BUFFER WITH SPACES SO WE CAN PRINT SHORT TEXT STRINGS WITH NO GARBAGE

FILL   LI R0,TXTBUF+1       START OF TEXT BUFFER (1ST BYTE IS BUFFER LENGTH).
LOOP3  MOVB @H80,*R0+       CO PY SPACE CHAR.
       CI R0,TXTBUF+41      DONE 40 CHARS?
       JNE LOOP3

*COPY TEXT FROM CALL LINK PARAMETER 1 TO TEXT BUFFER.

       LI R0,>2800          SET 1ST BYTE IN TEXT BUFFER TO BUFFER LENGTH.
       MOVB R0,@TXTBUF

       CLR R0               NOT S TRING ARRAY.
       LI R1,1              1ST PARAMETER.
       LI R2,TXTBUF         ADDRESS OF BUFFER.
       BLWP @STRREF         GET STRING PARAMETER.

*GO THROUGH BUFFER AND ADD XB OFFSET OF >60 TO EACH CHAR.

       LI R0,TXTBUF+1       START OF TEXT BUFFER (1ST BYTE IS BUFFER LENGTH).
LOOP2  AB @H60,*R0+         ADD OFFSET.
       CI R0,TXTBUF+41      DONE 40 CHARS?
       JNE LOOP2

*COPY TEXT TO VDP.

       MOV @CURRLN,R0       PRINT POSITION.
       LI R1,TXTBUF+1       ADDRESS OF 1ST CHAR IN BUFFER.
       LI R2,40             40 CHARS TO COPY.
       BLWP @VMBW

       AI R0,40             INCREMENT PRINT POSITION TO NEXT LINE.
       MOV R0,@CURRLN       SAVE PRINT POSITION.

*RETURN TO XB.

       CLR R0               CLEAR STATUS.
       MOVB R0,@STATUS      SEE E/A MANUAL SE CTION 24.11.2.
       B *R11


       END
```

Supporting the **TI-99/4** and **/4A**, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.

```
2260 !INSTRUCTION FORMAT 3
2270 !AND
2280 !INSTRUCTION FORMAT 4
2290 !AND
2300 !INSTRUCTION FORMAT 9
2310 !*********************
2320 !
2330 TS$=SEG$(ADDR_DATA_BIN$
,11,2)
2340 S$=SEG$(ADDR_DATA_BIN$
,13,4)
2350 D$=SEG$(ADDR_DATA_BIN$
,7,4)
2360 !PROCESS 'S' PART OF OP
CODE
2370 GOSUB 3250
2380 !PROCESS 'D' PART OF OP
CODE
2390 CALL CONV_BASE(D$,0,2,""
,REG,10)
2400 OP_TMP$=SEG$(INSTR_PR$
,1,4)
2410 IF OP_TMP$="LDCR" THEN
INSTR_PR$=INSTR_PR$&STR$(REG)
:: GOTO 2450
2420 IF OP_TMP$="STCR" THEN
INSTR_PR$=INSTR_PR$&STR$(REG)
:: GOTO 2450
2430 IF OP_TMP$="XOP " THEN
INSTR_PR$=INSTR_PR$&STR$(REG)
:: GOTO 2450
2440 INSTR_PR$=INSTR_PR$&"R"&
STR$(REG)
2450 GOTO 3070
2460 !
2470 !*********************
2480 !INSTRUCTION FORMAT 5
2490 !*********************
2500 !
2510 C$=SEG$(ADDR_DATA_BIN$
,9,4)
2520 CALL CONV_BASE(C$,0,2,""
,C,10)
2530 W$=SEG$(ADDR_DATA_BIN$
,13,4)
2540 CALL CONV_BASE(W$,0,2,""
,W,10)
2550 INSTR_PR$=INSTR_PR$&"R"&
STR$(W)&","&STR$(C)
2560 GOTO 3070
2570 !
2580 !*********************
2590 !INSTRUCTION FORMAT 6
2600 !*********************
2610 !
2620 !PROCESS THE 'S' PART OF
THE OP CODE USING THE CODE FOR
```

```
THE 'D' PART AS THIS IS THE
ONLY OPERAND AND WE DON'T WANT
A COMMA AFTER IT
2630 TD$=SEG$(ADDR_DATA_BIN$
,11,2)
2640 D$=SEG$(ADDR_DATA_BIN$
,13,4)
2650 GOSUB 3500
2660 GOTO 3070
2670 !
2680 !*********************
2690 !INSTRUCTION FORMAT 7
2700 !*********************
2710 !
2720 !NOTHING FURTHER TO ADD
2730 !FOR THIS INSTRUCTION
2740 !FORMAT
2750 GOTO 3070
2760 !
2770 !*********************
2780 !INSTRUCTION FORMAT 8
2790 !*********************
2800 !
2810 OP_TMP$=SEG$(INSTR_PR$
,1,4)
2820 IF OP_TMP$="LIMI" THEN
GOTO 2930
2830 IF OP_TMP$="LWPI" THEN
GOTO 2930
2840 IF OP_TMP$="STST" THEN
GOTO 2980
2850 IF OP_TMP$="STWP" THEN
GOTO 2980
2860 !AI, ANDI, CI, LI, ORI OP
CODES
2870 W$=SEG$(ADDR_DATA_BIN$
,13,4)
2880 CALL CONV_BASE(W$,0,2
,"",REG,10) :: INSTR_PR$=
INSTR_PR$&"R"&STR$(REG)&","
2890 CALL READ_WORD(ADDR,BYTE1
,BYTE2,ADDR_DATA) :: CALL
CONV_BASE("",ADDR_DATA,10,
OPERAND$,TMP,16) :: INSTR_PR$=
INSTR_PR$&">"&OPERAND$
2900 TS_WORD_PR$=OPERAND$
2910 CALL CONV_ASCII(
TS_ASCII_PR$,BYTE1,BYTE2)
2920 GOTO 3070
2930 !LIMI AND LWPI OP CODES
2940 CALL READ_WORD(ADDR
,BYTE1,BYTE2,ADDR_DATA) ::
CALL CONV_BASE("",ADDR_DATA
,10,OPERAND$,TMP,16) ::
INSTR_PR$=INSTR_PR$&
">"&OPERAND$
2950 TS_WORD_PR$=OPERAND$
```

```
2960 CALL CONV_ASCII(
TS_ASCII_PR$, BYTE1, BYTE2)
2970 GOTO 3070
2980 !STST AND STWP OP CODES
2990 W$=SEG$(ADDR_DATA_BIN$
, 13, 4)
3000 CALL CONV_BASE(W$, O, 2
, "", REG, 1O):: INSTR_PR$=
INSTR_PR$&" R" &STR$(REG)
3010 GOTO 3070
3020 !
3030 !* * * * * * * * * * * * * * * * * * * * * *
3040 !PRINT RESULTS
3050 !* * * * * * * * * * * * * * * * * * * * * *
3060 !
3070 CALL LINK("PRINT"
, ADDR_PR$&"   " &DATA_WORD_PR$&"
" &INSTR_PR$&RPT$(" ", 25-LEN
(INSTR_PR$))&DATA_ASCII_PR$)
3080 IF TS_WORD_PR$<>"" THEN
CALL LINK("PRINT"
, "        " &TS_WORD_PR$&
RPT$(" ", 26)&TS_ASCII_PR$)
3090 IF TD_WORD_PR$<>"" THEN
CALL LINK("PRINT"
, "        " &TD_WORD_PR$&
RPT$(" ", 26)&TD_ASCII_PR$)
3100 !
3110 !* * * * * * * * * * * * * * * * * * * * * *
3120 !CHECK IF KEY IS PRESSED
AND RETURN TO START ADDRESS
PROMPT IF SO
3130 !* * * * * * * * * * * * * * * * * * * * * *
3140 !
3150 CALL KEY(O, K, S)
3160 IF S<>O THEN CALL LINK(
"GRPHIC") :: GOTO 1230
3170 GOTO 1420
3180 !
3190 !* * * * * * * * * * * * * * * * * * * * * *
3200 !SUBROUTINE:
3210 !PROCESS 'S' PART OF OP
3220 !CODE
3230 !* * * * * * * * * * * * * * * * * * * * * *
3240 !
3250 IF TS$=" OO" THEN 3300

3260 IF TS$=" O1" THEN 332O
3270 IF TS$=" 1O" AND S$=" OOOO"
THEN GOTO 3340
3280    IF    TS$=" 1O"    AND
S$<>" OOOO" THEN GOTO 3370
3290 GOTO 341O :: !TS$=" 11"
3300 CALL CONV_BASE(S$, O, 2
, "", REG, 1O) :: INSTR_PR$=
INSTR_PR$&" R" &STR$(REG)&", "
3310 RETURN
3320 CALL CONV_BASE(S$, O, 2
```

```
, "", REG, 1O) :: INSTR_PR$=
INSTR_PR$&" * R" &STR$(REG)&", "
3330 RETURN
3340 CALL READ_WORD(ADDR, BYTE1
, BYTE2, ADDR_DATA)     ::    CALL
CONV_BASE(" ", ADDR_DATA, 1O
, SYMBOLIC$, TMP, 16) ::
INSTR_PR$=INSTR_PR$&" @>" &
SYMBOLIC$&", "
3350  TS_WORD_PR$=SYMBOLIC$  ::
CALL CONV_ASCII(TS_ASCII_PR$
, BYTE1, BYTE2)
3360 RETURN
3370 CALL READ_WORD(ADDR, BYTE1
, BYTE2, ADDR_DATA)     ::    CALL
CONV_BASE(" ", ADDR_DATA, 1O
, INDEXED$, TMP, 16)     ::    CALL
CONV_BASE(S$, O, 2, "", REG, 1O)
3380 INSTR_PR$=INSTR_PR$&
" @>" &INDEXED$&" (R" &STR$(REG)&
"), "
3390  TS_WORD_PR$=INDEXED$  ::
CALL CONV_ASCII(TS_ASCII_PR$
, BYTE1, BYTE2)
3400 RETURN
3410 CALL CONV_BASE(S$, O, 2, ""
, REG, 1O) :: INSTR_PR$=
INSTR_PR$&" * R" &STR$(REG)&
" +, "
3420 RETURN
3430 !
3440 !* * * * * * * * * * * * * * * * * * * * * *
3450 !SUBROUTINE:
3460 !PROCESS 'D' PART OF OP
3470 !CODE
3480 !* * * * * * * * * * * * * * * * * * * * * *
3490 !
3500 IF TD$=" OO" THEN 355O
3510 IF TD$=" O1" THEN 357O
3520 IF TD$=" 1O" AND D$=" OOOO"
THEN GOTO 3590
3530    IF    TD$=" 1O"    AND
D$<>" OOOO" THEN GOTO 3620
3540 GOTO 3660 :: !TD$=" 11"
3550 CALL CONV_BASE(D$, O, 2, ""
, REG, 1O) :: INSTR_PR$=
INSTR_PR$&" R" &STR$(REG)
3560 RETURN
3570 CALL CONV_BASE(D$, O, 2, ""
, REG, 1O) :: INSTR_PR$=
INSTR_PR$&" * R" &STR$(REG)
3580 RETURN
3590 CALL READ_WORD(ADDR, BYTE1
, BYTE2, ADDR_DATA)     ::    CALL
CONV_BASE(" ", ADDR_DATA, 1O
, SYMBOLIC$, TMP, 16)::INSTR_PR$=
INSTR_PR$&" @>" &SYMBOLIC$
```

```
3600 TD_WORD_PR$=SYMBOLIC$ ::
CALL CONV_ASCII(TD_ASCII_PR$
,BYTE1,BYTE2)
3610 RETURN
3620 CALL READ_WORD(ADDR,BYTE1
,BYTE2,ADDR_DATA)    ::    CALL
CONV_BASE("",ADDR_DATA,10
,INDEXED$,TMP,16)    ::    CALL
CONV_BASE(D$,0,2,"",REG,10)
3630 INSTR_PR$=INSTR_PR$&
"@>"&INDEXED$&"(R"&STR$(REG)&
")"
3640 TD_WORD_PR$=INDEXED$ ::
CALL CONV_ASCII(TD_ASCII_PR$
,BYTE1,BYTE2)
3650 RETURN
3660 CALL CONV_BASE(D$,0,2
,"",REG,10) :: INSTR_PR$=
INSTR_PR$&"*R"&STR$(REG)&"+"
3670 RETURN
3680 !
3690 !*********************
3700 !SUBPROGRAM:
3710 !CONVERT NUMBER BASES
3720 !
3730 !IN$=INPUT NUMBER AS STRING
3740 !IN=INPUT NUMBER IN DECIMAL
(0 FOR OTHER INPUT BASES)
3750 !IN_BASE=INPUT BASE: 2, 10,
16
3760 !OUT$=OUTPUT NUMBER AS STRING
3770 !OUT=OUTPUT NUMBER IN DECIMAL
('TMP' FOR OTHER OUTPUT BASES)
3780    !OUT_BASE=OUTPUT    BASE
REQUIRED: 2, 10, 16
3790 !*********************
3800 !
3810 SUB CONV_BASE(
IN$,IN,IN_BASE,OUT$,OUT
,OUT_BASE)
3820 !**CONVERT INPUT NUMBER TO
DECIMAL (IF NOT ALREADY)**
3830    IF    IN_BASE=10    THEN
DECIMAL=IN :: GOTO 3980
3840 !IF INPUT NUMBER IS HEX,
CHANGE HEX DIGITS A-F (CHAR 65-70)
TO
CHARS    58-63,    THEN    WORK    OFF
(ASC(DIGIT)-48)    AS    THE    DIGITS
VALUE
3850 IF IN_BASE<>16 THEN 3910
3860 FOR LOOP=1 TO LEN(IN$)
3870 DIG$=SEG$(IN$,LOOP,1)
3880    IF    DIG$>="A"    THEN
IN$=SEG$(IN$,1,LOOP-)&
CHR$(ASC(DIG$)-7)&SEG$(
IN$,LOOP+1,LEN(IN$))
3890 NEXT LOOP
3900 !CONVERT TO DECIMAL
3910 MULT=1
3920 DECIMAL=0
3930    FOR    LOOP=LEN(IN$)TO    1
STEP-1
3940 DECIMAL=DECIMAL+((ASC(
SEG$(IN$,LOOP,1))-48)*MULT)
3950 MULT=MULT*IN_BASE
3960 NEXT LOOP
3970 !**CONVERT DECIMAL NUMBER TO
REQUIRED OUTPUT BASE (IF OUTPUT
NOT DECIMAL)**
3980    IF    OUT_BASE=10    THEN
OUT=DECIMAL :: GOTO 4130
3990    !SET    VALUE    OF    MOST
SIGNIFICANT DIGIT, ASSUMING A MAX
VALUE OF 65535
4000    IF    OUT_BASE=2    THEN
MULT=32768 ELSE IF OUT_BASE=
10    THEN    MULT=10000    ELSE    IF
OUT_BASE=16    THEN    MULT=4096
ELSE PRINT "OUTPUT BASE NOT
VALID" :: STOP
4010 OUT$=""
4020 DIVIDE=INT(DECIMAL/MULT)
4030 OUT$=OUT$&CHR$(DIVIDE+48)
4040 DECIMAL=DECIMAL-(DIVIDE*
MULT)
4050 MULT=MULT/OUT_BASE
4060 IF MULT>=1 THEN GOTO 4020
4070 !CONVERT HEX DIGITS BACK
4080 IF OUT_BASE<>16 THEN
GOTO 4130
4090 FOR LOOP=1 TO LEN(OUT$)
4100 DIG$=SEG$(OUT$,LOOP,1)
4110 IF DIG$>=":" THEN OUT$=
SEG$(OUT$,1,LOOP-1)&CHR$(
ASC(DIG$)+7) &SEG$(OUT$
,LOOP+1,LEN(OUT$))
4120 NEXT LOOP
4130 SUBEND
4140 !
4150 !*********************
4160 !SUBPROGRAM:
4170 !READ ONE WORD FROM
4180 !MEMORY
4190 !*********************
4200 !
4210 SUB READ_WORD(ADDR
,BYTE1,BYTE2,ADDR_DATA)
4220 IF ADDR<32768 THEN CALL
PEEK(ADDR,BYTE1,BYTE2)ELSE
CALL PEEK(ADDR-65536,BYTE1
,BYTE2)
4230 ADDR_DATA=(BYTE1*256)+
BYTE2
4240 ADDR=ADDR+2
4250 SUBEND
4260 !
```

```
4270 !*********************
4280 !SUBPROGRAM:
4290 !CONVERT BYTES TO ASCII
4300 !*********************
4310 !
4320 SUB CONV_ASCII(ASCII$
,BYTE1,BYTE2)
4330 IF(BYTE1<33 OR BYTE1>95)
THEN IF BYTE1=32 THEN ASCII$=
CHR$(96) ELSE ASCII$="." ELSE
ASCII$=CHR$(BYTE1)
4340 IF(BYTE2<33 OR BYTE2>95)
THEN IF BYTE2=32 THEN ASCII$=
ASCII$&CHR$(96)ELSE ASCII$=
ASCII$&"." ELSE ASCII$=ASCII$&
CHR$(BYTE2)
4350 ASCII$=ASCII$&"/"
```

```
4360 !ADJUST BYTES FOR >60 ASCII
OFFSET AND REPEAT
4370 BYTE1=BYTE1-96 ::
 BYTE2=BYTE2-96
4380 IF(BYTE1<33 OR BYTE1>95)
THEN IF BYTE1=32 THEN ASCII$=
ASCII$&CHR$(96)ELSE ASCII$=
ASCII$&"." ELSE ASCII$=
ASCII$&CHR$(BYTE1)
4390 IF(BYTE2<33 OR BYTE2>95)
THEN IF BYTE2=32 THEN ASCII$=
ASCII$&CHR$(96)ELSE ASCII$=
ASCII$&"." ELSE ASCII$=ASCII$&
CHR$(BYTE2)
4400 SUBEND
```

Guess Who

A program to type in that produces a picture of somebody on screen. I have no idea myself who it is, so please let us know when you've run it! J

Subject: [TI-99/4A] Guess who
From: kabldb <no_reply@yahoogroups.com>
Date: Sat, 29 Sep 2007 12:45:26 -0000
To: ti99-4a@yahoogroups.com

Now who could this be ?

```
1 CALL SCREEN(1)
2 CALL CLEAR
3 CALL CHAR(33,"")
4 CALL CHAR(34,"1E397E7F7E797452")
5 CALL CHAR(35,"008060308C42A15")
6 CALL CHAR(36,"000106081123458A")
7 CALL CHAR(37,"F81C7EFEDE962E4A")
8 CALL CHAR(38,"000000000000018")
9 CALL CHAR(39,"")
10 CALL CHAR(40,
"000000000003844")
11 CALL CHAR(41,"")
12 CALL CHAR(42,"000000010000001")
13 CALL CHAR(43,
"380700F807798703")
14 CALL CHAR(44,
"29249051 6ACCC488")
15 CALL CHAR(45,"2804F20D0301")
16 CALL CHAR(46,"14204FB0C080808")
17 CALL CHAR(47,
"9424098E57262B15")
18 CALL CHAR(48,"60831C60C7F840B")
19 CALL CHAR(49,
"00800001C101012")
20 CALL CHAR(50,
"828282040404009")
21 CALL CHAR(51,
"001C224141818202")
22 CALL CHAR(52,"")
23 CALL CHAR(53,
"070801839363F6A")
24 CALL CHAR(54,"489010202020A")
25 CALL CHAR(55,"")
26 CALL CHAR(56,"808080808080808")
27 CALL CHAR(57,
"140808080404040 7")
28 CALL CHAR(58,
"104838B45C3AEE56")
29 CALL CHAR(59,"020202030C10102")
30 CALL CHAR(60,
"090A1C826010 0808")
31 CALL CHAR(61,
"0408102078040442")
32 CALL CHAR(62,"")
33 CALL CHAR(63,
"5054A9AAA69282A2")
34 CALL CHAR(64,
"E0302C231010 0804")
35 CALL CHAR(65,
"00000FF1C0000304")
36 CALL CHAR(66,"808080FF9C80E01")
37 CALL CHAR(67,"060C35C40808102")
38 CALL CHAR(68,
"0A452DA5C5494343")
39 CALL CHAR(69,
"2023202010100808")
40 CALL CHAR(70,"0810E1214181")
41 CALL CHAR(71,"8282820204 0488F")
```

Supporting the **TI-99/4** and **/4A**, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.

```
42 CALL CHAR(72,"")
43 CALL CHAR(73,
"A96D7C2834281OFA")
44 CALL CHAR(74,"03008040300F00C")
45 CALL CHAR(75,"FC030102OCF")
46 CALL CHAR(76,"1FE08040201F")
47 CALL CHAR(77,"C00001020CF")
48 CALL CHAR(78,
"8B972EO28D76FC78")
49 CALL CHAR(79,
"1020408000000102")
50 CALL CHAR(80,"01OE1020408")
51 CALL CHAR(81,"")
52 CALL CHAR(82,"0304081020204O4")
53 CALL CHAR(83,"07030301O1")
54 CALL CHAR(84,"408C4FF807986")
55 CALL CHAR(85,
"OOOOOOOFOOCO2O1")
56 CALL CHAR(86,"0000000001")
57 CALL CHAR(87,
"OO1831FFAOOO82O1")
58 CALL CHAR(88,
"FO608000000038E")
59 CALL CHAR(89,"04081020408")
6O CALL CHAR(90,"")
61 CALL CHAR(91,"")
62 CALL CHAR(92,"8O8O404020100F")
63 CALL CHAR(93,
"OOOO102060AO100F")
64 CALL CHAR(94,"OOOOOOO2OOOCO4")
65 CALL CHAR(95,
"O100080484828282")
66 CALL CHAR(96,"OO808O404040408")
67 CALL CHAR(97,"")
68 CALL CHAR(98,
"FC702020101OO808")
69 CALL CHAR(99,"")
7O CALL CHAR(1OO,"")
71 CALL CHAR(1O1,"")
72 CALL CHAR(1O2,"")
73 CALL CHAR(1O3,"")
74 CALL CHAR(1O4,"4041221CO3")
75 CALL CHAR(1O5,"83O4083OC")
76 CALL CHAR(1O6,"")
77 CALL CHAR(1O7,"")
78 CALL CHAR(1O8,"O8O8O8")
79 CALL CHAR(1O9,"")
8O CALL CHAR(11O,"")
81 CALL CHAR(111,"")
82 CALL CLEAR
83 FOR L=O TO 7
84 FOR C=O TO 9
85 CALL HCHAR(L+7,C+12,32+L*10+C)
86 NEXT C
87 NEXT L
88 CALL SCREEN(15)
89 GOTO 89
```

A mystery program for Super Extended BASIC.

Don't worry if you haven't got the cartridge. This is the early 21st century. If you have Win99/4A then you have already got Super Extended BASIC!

```
Subject:   [TI-99/4A] For Super Extended BASIC
From:      ti99_forever <ti99_forever@yahoo.com>
Date:      Fri, 2 Nov 2007 15:45:59 -0700 (PDT)
To:        ti99-4a@yahoogroups.com
```

```
1 DATA O5, 9F, BF, DF, FF, E3, O1
6 DATA O9, 8E, O1, A4, O2, C5, O1
, 90, B6, D3, O6
11 DATA O3, 91, B7, D4, O5
16 DATA O3, 92, B8, D5, O4
21 DATA O5, A7, O4, 93, BO, D6, O5
26 DATA O3, 94, B1, D7, O6
31 DATA O3, 95, B2, D8, O7
36 DATA O5, CA, O2, 96, B3, DO, O6
41 DATA O3, 97, B4, D1, O5
46 DATA O3, 98, B5, D2, O4
51 DATA O5, 85, O3, 90, B6, D3, O5
56 DATA O3, 91, B7, D4, O6
61 DATA O3, 92, B8, D5, O7
66 DATA O5, A4, O2, 93, BO, D6, O6
71 DATA O3, 94, B1, D7, O5
76 DATA O3, 95, B2, D8, O4
81 DATA O5, C5, O1, 96, B3, DO, O5
86 DATA O3, 97, B4, D1, O6
91 DATA O3, 98, B5, D2, O7
96 DATA O3, 9F, BF, DF, O1
106 DATA OO, O6, 40
109 DATA END
110 HEX$="123456789ABCDEF"
120 DEF HEX2DEC(A$)=POS(HEX$,SEG$(
A$,1,1),1)*16+POS(HEX$,SEG$(A$,2,1
),1)
125 I=1600
130 READ A$ :: IF A$="END"
 THEN 200
140 X=HEX2DEC(A$):: CALL POKEV(I
,X) :: I=I+1 :: PRINT ".";::
GOTO 130
200 PRINT "STARTING!" :: CALL LOAD
(-31796,6,64)
220 CALL PEEK(-31747,A):: A=A OR 1
 :: CALL LOAD(-31747,A)
230 CALL LOAD(-31794,1)
240 GOTO 240
```

Forcing a garbage collection in Extended BASIC.

This might be useful in a big program if you're about to do some intensive processing or some nice animation. Especially for animation, you won't want the system to do a random garbage collection in the middle of your routine. If you use this, then you can force it before hand! J

```
Subject:    [TI-99/4A] CALL LINK forces garbage collect
From:       ti99_forever <ti99_forever@yahoo.com>
Date:       Tue, 21 Aug 2007 00:48:02 -0700 (PDT)
To:         swpb@yahoogroups.com, ti99-4a@yahoogroups.com

Found this in Tigercub, just like I thought. Tip # 65.8...

Thank you, Jim Peterson, and the Sydney User Group in Australia!
I believe I've tested this before and it worked. Will have to try again!

CALL LOAD(-31885,144,"",-31858,81,169,152,0).

I suppose this may be an "XB Only" thing, but I wonder what it is poking?
```

Converting a DV80 file of a listing back into an XB program

```
Subject:    [TI-99/4A] ASCII (DV/80) --> PROG Converter
From:       "Eric Bray" <ewbray2000@yahoo.com>
Date:       Wed, 21 Nov 2007 11:21:19 -0000
To:         ti99-4a@yahoogroups.com
```

A while back a user was asking about where they could get a hold of the program that ran on an "actual" TI99/4A that converts an ASCII text file (DV/80) into a TIXB program. I found the program searching through my disks last night and here is its listed format for that user to copy!

```
10 @=1 :: _=2 :: ON WARNING
NEXT :: DEF A(A$)=POS(A$,"
",@)-@ :: DISPLAY AT(_,@)
ERASE ALL:"ASCII FILE CONVE
RTER":RPT$("-",20): : : :
:"YOU WANT INSTRUCTIONS? Y/
N" :: GOSUB 160
20 DISPLAY AT(8,@): : : :
:"INPUT  FILE? DSK1." ::
ACCEPT AT(12,17)
SIZE(-12)BEEP: B$ :: DISPLAY
AT(14,@):"OUTPUT FILE? DSK1
." :: ACCEPT AT(14,17)
SIZE(-12)BEEP: C$

30 OPEN #@:"DSK"&B$, INPUT
:: OPEN #_:"DSK"&
C$,OUTPUT,VARIABLE 163 ::
CALL CLEAR :: PRINT "ONE MO
MENT...": : :
40 LINPUT #@: D$ :: IF D$=""
OR LEN(D$)<3 THEN 40
50 IF EOF(@)=[ THEN LINPUT
#@: E$ :: D=ASC(E$) :: IF
D<49 OR D>57 THEN
D$=D$&E$ :: GOTO 50 ELSE
GOSUB 80 :: D$=E$ ::
 GOTO 50
60 D$=E$ :: GOSUB 80 ::
PRINT #_: CHR$(255)&
```

```
CHR$(255) :: CLOSE #@ ::
CLOSE #_
70 PRINT: :"THE FILE: ";
SEG$(C$,3,LEN(C$)-_):"HAS
BEEN CREATED": :"DO ANOTH
ER FILE?" :: CALL HCHAR
(23,20,30) :: GOSUB 160
:: CALL HCHAR(23,20,B) ::
CALL CLEAR :: IF C=@ THEN
20 ELSE END
80 D$=SEG$(D$,@,162) ::
D=A(D$) :: F=INT(VAL(
SEG$(D$,@,D))/256) :: G=VAL(
SEG$(D$,@,D))-(F*256) ::
PRINT D$ :: D$=CHR$(F)&
CHR$(G)&SEG$(D$,D+@
,LEN(D$)-D)&CHR$([) ::
PRINT #_:D$ :: RETURN
90 CALL HCHAR(8,@,32,512)
:: DISPLAY AT(9,8):
"INSTRUCTIONS":TAB(8);"---
---------" :: FOR E=@ TO 6
:: FOR F=12 TO 21 STEP _
:: READ A$ :: DISPLAY AT(
F,@):A$ :: NEXT F :: GOSUB
180 :: CALL HCHAR(12,@
,32,416) :: NEXT E ::
RETURN
100 DATA THIS UTILITY WILL
READ ANY, ASCII TEXT FILE
AND TURN IT,INTO A MERGE
FILE READY TO,LOAD AND RUN
IN TI-BASIC.
110 DATA ANY TEXT FILE
STORED IN, DIS/VAR 80 FORMAT
CAN BE,USED AS THE INPUT
FILE.,THE INPUT FILE CAN BE
A FILE,RETRIEVED WITH THE
TE11;
```

```
120 DATA WRITTEN WITH THE
TI-WRITER; ETC.,IF THE
INPUT FILE IS WRITTEN,USING
THE TI-WRITER YOU MUST,
REMOVE ALL<CR>*BEFORE*
,USING THIS PROGRAM.,""
130 DATA IF THE INPUT FILE
CONTAINS,LINES OF LONGER
THAN 162,"CHARACTERS, ANY
EXCESS WILL",BE DISCARDED..
,"",AFTER THE PROGRAM HAS
140 DATA CREATED THE MERGE
FILE YOU,MUST LOAD THE
MERGE FILE AND,REMOVE THE
ONE EXTRA SPACE,AT THE
START OF EACH LINE,BEFORE
IT WILL RUN.
150 DATA THEN SAVE THE
PROGRAM.,"","","",""
160 CALL SOUND(150,1100,[)
170 CALL KEY([,B,D) :: IF
B=78 THEN RETURN ELSE IF
B=89 THEN GOSUB 90 ::
DISPLAY AT(9,@): : : ::
RETURN ELSE 170
180 DISPLAY AT(24,_):"PRESS
ANY KEY TO CONTINUE" ::
CALL SOUND(150,1200,[)
190 CALL KEY([,D,F) :: IF
F=[ THEN 190 ELSE RETURN
```

Merry Christmas to all our readers.

An Oldie music program.

Subject: [TI-99/4A] Another oldie ...
From: kabldb <no_reply@yahoogroups.com>
Date: Sat, 27 Oct 2007 09:38:02 -0000
To: ti99-4a@yahoogroups.com

He he, can you name the song ?
Any comments on the code ? **J**

```
100 MU$=" E1D1C1D1E1E1E2D1D1D2E1G1G2E1D1C1
D1E1E1E1E1D1D1E1D1C4"
110 FR(0)=440
120 FR(1)=494
130 FR(2)=523
140 FR(3)=687
150 FR(4)=659
160 FR(5)=698
170 FR(6)=784
180 FOR LO=1 TO LEN(MU$)-1 STEP 2
190 NO=ASC(SEG$(MU$,LO,1))-67
200 LE=(ASC(SEG$(MU$,LO+1,1))-48)*250
210 CALL SOUND(LE,FR(NO),0,FR(NO)*1.01,0)
220 NEXT LO
```

2-Column Catalog

Subject: [TI-99/4A] 2-Column Catalog
From: "computerclassics2000" <computerclassics@hotmail.com>
Date: Fri, 12 Oct 2007 23:45:17 -0000
To: ti99-4a@yahoogroups.com

Here's one that I didn't remember writing, that I ran across in a September 1986 issue of the South Mobile and Alabama UG Newsletter.

LIST DISK CONTENTS FAST
The following four-step XBASIC program by Bill Gaskill of Colorado quickly provides a two-column display of the program and file names of your disk. It doesn't waste time listing seldom-used data such as file type, length and protection status. It does however show the disk name and number of used and unused sectors at the top of the listing. As written it searches only DSK1. Gaskill suggests saving it under a LOAD filename so you will be able to see what's on the disk in a matter of seconds.

```
1 @=1 :: OPEN #@:" DSK1.", INP
UT, RELATIVE, INTERNAL :: INP
UT #@:F$,E,E,F :: DISPLAY AT
(2,@)ERASE ALL:F$:" FREE=";F:
" USED=";E-F :: R=5 :: C=@
2 FOR H=@ TO 127 :: INPUT #@
:F$,D,E,F :: DISPLAY AT(R,C)
:F$ :: R=R+@ :: IF ABS(D)=0
THEN CLOSE #@ :: END
3 IF R<24 THEN 4 :: C=16 ::
R=5
4 NEXT H
```

XB Garbage Collection explained by Thierry Nouspikel

```
Subject:      [TI-99/4A] Re: CALL LINK forces garbage collect
From:         "Thierry Nouspikel" <nouspikel@yahoo.com>
Date:         Tue, 21 Aug 2007 09:27:29 -0000
To:           ti99-4a@yahoogroups.com
```

> > CALL LOAD(-31885,144,"",-31858,81,169,152,0).
> >
> > I suppose this may be an "XB Only" thing, but I wonder what it is poking?

It places >51A9, >9800 in the scratch-pad memory at addresses >838E-8391 and it sets the subroutine stack pointer (in >8373) to point at it.

This should cause the Basic (and presumably XB) interpreter to branch to >51A9, using >9800 as a GROM port, when returning from the current subroutine.

> >51A9 is the entry point of the garbage collection routine in the console GROMs.

Hope this helps,

      Thierry

Bruce Harrison's recipes

As it's Christmas, part of the enjoyment of it is nice food, besides nice presents and the Sound of Music on T.V. twice an hour (according to Monty Python!)
As another tribute to Bruce Harrison, here are a couple of his own recipes to wet our taste buds this Christmas.
In Bruce's own words…

Here are the major Features of these recipes:
They are guaranteed to be unhealthy.*
Using them will increase your risk of Heart Attacks.*
Each will greatly overfeed four people, except where otherwise noted.
Most who've tried them found them delicious.
With generous salt, each has very high sodium content.

* Anbody who has doubts about these statements should know that the author of these recipes has in fact suffered a heart attack.

Harrison's Home-made Hamburger Helper

You'll need: one large white onion or two small yellow ones
          1 1/2 lbs very lean hamburger
          12 ounces elbow macaroni
          1 large can crushed tomatoes
          2 tablespoons butter or margarine
          Salt (to taste)
          2 tablespoons A-1 sauce or Worcestershire sauce (optional)

Procedure: Start water to boiling in a 3-quart saucepan. Add some salt to this water. When it's boiling, add the elbow macaroni. Stir occasionally to keep macaroni from sticking to itself. Chop the onion into small pieces (not minced, just chopped). In a large skillet (10 inch or bigger) melt the butter or margarine (I use Soft Parkay) til it's bubbling. Put the chopped onion in the skillet. Shake in some salt. Stir-fry (or is that Saute?) the onions until they're starting to brown just a bit. Add the hamburger and thoroughly brown that, breaking it up while it browns.

Add the crushed tomatoes to the skillet, stir into the mixture thoroughly. Cover and reduce heat to a simmer. Let this simmer while the macaroni is boiling. When the macaroni is done, drain it thoroughly, then add to the skillet and stir completely into the mixture. It's ready!

Usually, we serve this along with some kind of vegetable. Goes nicely with string beans, peas, or even corn.

Variations: Other forms of pasta, (e.g. rotini or wide egg noodles) can be used. Heinz ready-made gravy (two jars) can be substituted for the tomatoes. Garlic powder or fresh garlic can be added if desired. The A-1 or Worcestershire can be left out. The gravy/onion mixture can be served over mashed potatoes instead of with pasta. Finally, a different meat can be used in place of hamburger. We've used ground turkey, for example, with excellent results, although this makes a far less unhealthy dish because of the low-fat quality of the ground turkey.

Scottish style Split Pea Soup

You'll need: 2 packages dried split peas (green or yellow)
          Dried chopped onions (not minced ones)
          2 large or three small ham hocks
          coarse ground black pepper

Procedure: Not a "quickie". In a large pot (preferably a 5-quart dutch oven type) place the dried split peas, then cover with water to nearly fill the pot. Set this aside (covered) over night to soften the peas. Next morning, put in the ham hocks and onions, cover, place over the fire, and get it boiling vigorously. This may boil over easily, so keep an eye on it and reduce heat when necessary. Let it go at a rapid simmer for about three hours. Add some black pepper if desired, but not too much. Pull out the ham hocks. Cut away the rinds, cut the meat off the bones, chop that in small pieces and return the meat to the pot. Simmer until it's well thickened. It's done!

Variations: Some people prefer regular ham to the hocks. Some people like to add some finely shredded carrots, this adds color, but doesn't change the taste. This same recipe can be used with dried lentils instead of peas, but then the carrots are a "must".

Scottish Potted Meat

You'll need: 2 1/2 to 3 lbs shin beef (aka shank beef)
          2 1/2 to 3 lbs ox tails
          Salt
          Patience galore

Procedure: This dish was handed down to me by my grandmother. It is simple, uses very cheap meats, and people either love it or hate it. In any case it takes lots of time to prepare and must afterward be refrigerated at least overnight. Not a quickie! Use a large pot, peferably a thick walled dutch oven type with cover. Put the meat, complete with its bones, into the pot first. Add enough water to more than cover the meat, and some salt. Put this over high heat until it boils rapidly, then lower the heat to a moderate simmer. Cover. Wait about three hours for this to properly simmer all the meat. Some water may have to be added now and then, but just enough to keep the meat submerged. Remove from heat. Take the meats out of the water and let them cool a bit. Carefully remove the ox tail meat from its bones. Toss the fat aside. Shred the ox tail meat and the shin beef by hand into small pieces of irregular shape. Add the meat back into the liquid in the pot. Spoon or ladle the mixture into bowls or cups, being sure each has about the same proportion of meat and juice. Put these bowls, cups or other containers (preferably covered with something) into the refrigerator and leave them there for at least eight hours. It's done! Each cup or bowl will be filled with a gelatin that's chock full of meat, and each will have a layer of suet that's floated to the top. Peel off this suet layer and discard. Serve sliced up cold. The eater can put some ketchup, horseradish or other condiment on to his or her taste. See, Barry, that's two dishes with no onions.

Variations: None, for all practical purposes. In Scotland, this was made with the meat from a cow's cheek, (the facial ones) but that's hard to find in this country. That's why it is sometimes called Potted Head. The shin and ox tail serve as a good substitute, because both of those create a nice gelatin. Don't try this recipe with more expensive kinds of meat, as those won't "gel" correctly.

Quick and Easy Chili

You'll need: 1/2 pound lean ground beef
                Two 15-oz cans Dark Red kidney beans
                One large can crushed tomatoes
                Dried Chopped onions
                Chili powder
                Coarse ground black pepper
                Olive oil (one tablespoon)

Procedure: Pre-heat the olive oil in a skillet. Add the ground beef in chunks, stir to brown all over. Add about a tablespoon of dried chopped onions, brown lightly. Pour into a three quart saucepan. Add the kidney beans and tomatoes, stir. Add about 3/4 teaspoon of chili powder, half a teaspoon of the black pepper, stir. Bring the mixture to a boil, then reduce heat and simmer uncovered for about twenty minutes. Serve in bowls.

Variations: This makes a VERY mild chili that's ready quickly. Obviously you may want a spicier kind, so you could add more chili powder, or some Tabasco, cayenne pepper, or even dry mustard for additional zip. Some like to serve with shredded cheddar on top, and chopped raw onions. In effect, you should consider this a "starting point" recipe from which you can develop your own favorite.

Our gratitude to Bruce Harrison for his recipes and all the software he provided us with. He wrote nearly as many recipes as he did TI programs, so there are lots more of his recipes to choose from. Maybe I'll include some more of his recipes sometime? Maybe next Christmas **J**

That's all from me now. I'm signing off for another year and I am now packing away all my bits and bobs ready for moving house five days before Christmas!

A special thank you to Trevor by printing the magazines and posting them out for me this time around!

Merry Christmas

THE END
BUT...
RICHARD TWYNING WILL RETURN

Kids Still say the Darnest Things!

From: ERIC BRAY <ewbray2000@yahoo.com>
Date: Mon, 3 Dec 2007 15:53:54 -0800 (PST)
To: TI99/4A <ti99-4a@yahoogroups.com>

A 1st grade school teacher had twenty-six students in her class. She presented each child in her classroom the 1st half of a well-known proverb and asked them to come up with the remainder of the proverb.

The kids text is in bold.

It's hard to believe these were actually done by first graders. Their insight may surprise you. While reading, keep in mind that these are first-graders, 6-year-olds, because the last one is a classic**!**

1. Don't change horses **until they stop running**.
2. Strike while the **bug is close**.
3. It's always darkest before **Daylight Saving Time**.
4. Never underestimate the power of **termites**.
5. You can lead a horse to water but **How?**
6. Don't bite the hand that **looks dirty**.
7. No news is **impossible**
8. A miss is as good as a **Mr.**
9. You can't teach an old dog new **Math**
10. If you lie down with dogs, you'll **stink in the morning**.
11. Love all, trust **Me.**

Supporting the **TI-99/4** and /4A, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.

12. The pen is mightier than the **pigs**.
13. An idle mind is **the best way to relax**.
14. Where there's smoke there's **pollution**.
15. Happy the bride who **gets all the presents**.
16. A penny saved is **not much**.
17. Two's company, three's **the Musketeers**.
18. Don't put off till tomorrow what **you put on to go to bed**.
19. Laugh and the whole world laughs with you, cry and **You have to blow your nose**.
20. There are none so blind as **Stevie Wonder**.
21. Children should be seen and not **spanked or grounded**.
22. If at first you don't succeed **get new batteries**.
23. You get out of something only what you **See in the picture on the box**
24. When the blind lead the blind **get out of the way**.
25. A bird in the hand **is going to poop on you**.

And the WINNER and last one!
26. Better late than **Pregnant**

Francesco Lama – Media Librarian

Many thanks to Francesco for supplying us with a new updated module library listing.

MODULE LIBRARY (Updated 8th October)

```
    TITLE                          QTY IN STOCK           PRICE (POUNDS)

32k SUPERSPACE (MODIFIED ROMOX) ........ 1 ................... 25.00

ADDITION & SUBTRACTION 1 ............... 1 ...................  3.00
ADVENTURE + PIRATE TAPE (OTHERS TOO) ... 5 ...................  5.00
ADVENTURE MODULE ONLY .................. 6 ...................  3.50
ALPINER ................................ 3 ...................  8.00
A-MAZING ...............................10 ...................  3.00

BEGINNING GRAMMAR ...................... 4 ...................  3.00
BLASTO ................................. 1 ...................  5.00

CAR WARS ............................... 1 ...................  4.00
CHISHOLM TRAIL ......................... 1 ...................  3.50
CONNECT FOUR ........................... 1 ...................  3.50

DISK MANAGER ........................... 4 ...................  2.00
DISK MANAGER 2 ......................... 3 ...................  4.50
DIVISION 1 ............................. 1 ...................  3.00

EDITOR ASSEMBLER + MANUALS & DISKS ..... 4 ................... 25.00
EXTENDED BASIC STILL IN ORIG. BOX ...... 1 ................... 25.00
EXTENDED BASIC + MANUAL ................ 4 ................... 22.50
EXTENDED BASIC MODULE .................. 6 ................... 15.00

HOUSEHOLD BUDGET MANAGEMENT ............ 4 ...................  3.50
HUNT THE WUMPUS ........................ 1 ...................  4.00
HUSTLE (EA VERSION ONLY) ............... .. ...................  2.00

INDOOR SOCCER .......................... 2 ...................  4.00
JAWBREAKER 2 ........................... 1 ...................  4.00
```

Supporting the **TI-99/4** and **/4A**, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.

MODULE LIBRARY continued…

```
MINI MEMORY + LINE BY LINE ASS. ........ 4 ................... 15.00
MINI MEMORY AS ABOVE + MINI WRITER ..... 2 ................... 18.00
MULTIPLAN + SOFTWARE + MANUAL .......... 2 ................... 30.00
MULTIPLICATION 1 ....................... 1 ...................  3.00
MUNCHMAN ............................... 1 ...................  3.50
MUSIC MAKER ............................ 1 ...................  3.00


PARSEC ................................. 1 ...................  4.00
PERSONAL RECORD KEEPING ................ 4 ...................  3.50
PERSONAL REPORT GENERATOR .............. 1 ...................  5.50
PHYSICAL FITNESS ....................... 1 ...................  4.00
PROTECTOR (NOT MK 2 CONSOLES) .......... 3 ...................  5.00
PROTYPER ............................... 1 ................... 20.00


SHAMUS (NOT MK 2 CONSOLES) ............. 2 ...................  3.50
SPEECH EDITOR .......................... 1 ...................  3.50
SUPER DEMON ATTACK ..................... 1 ...................  3.00
SUPER EXTENDED BASIC ................... 1 ................... 30.00


TERMINAL EMULATOR 2 .................... 6 ...................  5.00
THE ATTACK ............................. 1 ...................  4.00
TI INVADERS ............................ 1 ...................  4.00
TI LOGO + FOLDER & MANUAL .............. 1 ................... 15.00
TI WRITER MODULE ....................... 3 ...................  8.00
TOMBSTONE CITY ......................... 3 ...................  4.00


VIDEO GAMES 1 .......................... 1 ...................  3.50


YAHTZEE ................................ 2 ...................  3.00
```

The 22nd TI Treffen – By Berry Harmsen

Dear TI Friends

The last international guests just left Schiphol Airport. The 22nd European TI Treff is over. And we had a lot of fun and... a lot new items for our TI hobby. We had about 50 visitors from seven countries. I will write a full report in the next days.
These are the main items of Hilversum:

   * **Fred Kaal** (TIGG) demonstrated his latest software.   * **Richard Twyning** (TIUGUK) showed new possibilities to connect the TI with Bluetooth. * **Michael Becker** (SNUG) released new hardware for 80 columns.   * There was new software from TI Club Errorfree Germany demonstrated by **Wolfgang Bertsch**.

The next Treff (2008) will be again in Germany. There are two options: Paderborn or Augsburg. In 2009 the European TI Users will go to Vienna in Austria. Then in 2010 we will celebrate 25th birthday of the TI Treffen. That Treff will be organized by the TI Users Group of the UK.

We had also interesting demo's about Fortran, internet file handling and the use of Compact Flash card on the TI99. I will give full details to the visitors of the Chicago Faire on October, 20th in Evanston. I hope to see a lot of you there. Don't miss it!!

Groeten van Berry Harmsen

Supporting the **TI-99/4** and /4A, the MYARC GENEVE 9640, Michael Becker SGCPU card,
And any other compatible machine.