User Manual

for

GeekKlok

Documentation release version 2.6-BU

written by Raymond Weisling

May 2006

PT Zetalink Technology Indonesia
FAX (USA) + 1–708–575–6950

Kaliurang Pratama E3
http://zetalink.biz

Yogyakarta,
sales-_-@zetalink.biz

The GeekKlok

SCOPE

This manual is intended for assembled versions of the GeekKlok.

HISTORY

The GeekKlok is an outgrowth of the Four Letter Word (FLW), also designed by Raymond Weisling and produced by Zetalink. While a word sculpture may be fun for some people, some others just don't "get it". Clocks, on the other hand, have a much more universal appeal. Inspired by the dynamic nature of the FLW in its hangman game modes, the GeekKlok is an attempt to make clock-watching much more fun. It may well bring out the Hidden Geek in all of us.

The GeekKlok is built on the same printed circuit board as the FLW, but there are numerous features that set it apart from the FLW. It has a clock and calendar chip with battery backup; this chip also has a nonvolatile memory for storage of user settings.

OPERATION

When the unit is first turned on it will always display a greeting message: "GEEK KLOK BY RAYMOND WEISLING" followed by the internal program version number, and then the word OK. At this point it will begin displaying the time. The very first time it is turned on the time might not be not set, so basic settings may be required. See the section on SETTING TIME.

Once the time and other settings have been completed, the battery-operated Real-Time Clock (RTC) chip will keep this information. If, however, the battery is removed, then the setting are lost and will have to be set again. There are default settings that are automatically placed into the RTC backup memory if it is found to be empty.

There are three groups of user settings, operated by the three buttons, SET, ADV and OPT. The first is the TIME setting editor/browser, which only sets hours and minutes. When exiting this setting, if the minutes have been changed, the seconds are automatically cleared to zero.

Another user preference editor/browser is for a group of data registers. These registers contain information for the calendar, automatic daylight savings time (summer time) corrections, display blanking hours, font change timings, date display program selection, etc. For further information, refer to the section on REGISTER SETTINGS

The third user preference editor/browser is for selecting fonts. Fifty numeric fonts are provided for variety and novelty use. One or more fonts may be selected to appear all of the time or to cycle with different durations. For further information, see the section on FONT SETTINGS.

SETTING TIME

Pressing the SET button one time will show the prompt **TIME** on the display. Pressing the ADV button once will enter the hours setting mode. The ADV button can be pressed once to advance the hours, or held down to automatically advance. If the OPT button is held down when ADV is pressed, the hours decrease.

Pressing the SET button a second time will enter the minutes setting mode. Just as with hours, ADV steps through the minutes, and the OPT button pressed when pressing ADV will reverse the direction. Within the menus the OPT button always reverses the step-wise direction of the other buttons.

Pressing SET again moves back to hours. SET always alternates between minutes and hours. To exit the time setting mode, press SET and then ADV together at the same time. This will return the clock to the normal time display, using the last time entered, and with seconds set to zero. In the various editing modes, the display always uses the font called Normal (NOR).

These are the three setting buttons as seen from the rear of the clock.

REGISTER SETTINGS

A number of user preference registers is provided to allow you to customize your GeekKlok. Registers contain numerical values between two preset limits, appropriate to their individual function. The display shows a two-letter register name and one or two digits representing the current value. You can browse through the registers and change them at any time.

To enter the register browser/editor, press SET once (to display the **TIME** prompt) and then a second time. The prompt will show **REGS** on the display. Pressing ADV will then enter the register editor. Note that some register values change under control of the clock after you set them. For example, the date, day of week, month and year will change to reflect the current calendar changes. Values that remain fixed to your setting are noted below as **FIXED**. Pressing SET advances to the next register. The OPT button will reverse its normal stepwise direction of either SET or ADV. Holding either button will auto-step in the direction presently set. When setting a register (or font), it is necessary to step to the next item with SET before pressing SET and ADV at the same time to exit the menu.

Note

Regardless of the setting for 12- or 24-hour display modes, all register settings that involve specifying hours use the 24-hour time convention. This is done to simplify the display, since no additional characters are available for a.m. or p.m. indications.

FP 1 – 99 (Font, Primary) This value determines the duration the primary font is used to show the time (hours and minutes). *It represents tens of seconds*, so a value of 6 means that the primary font will be shown for sixty seconds. The minimum duration is 10 seconds and the maximum duration is 990 seconds. It is possible to have the primary font show at all times by setting the FS register to zero. Any one of the 50 fonts (or two animation pseudo-fonts) may be used as the primary font. These are set using the font group browser/editor. **FIXED**

FS 0 – 99 (Font, Secondary) This value determines the duration the secondary font is used to show the time. Primary and secondary fonts alternate, based on the durations in these two registers.The unit is one second, so a value of 99 means that the secondary font is visible for 99 seconds maximum. Any one of the 50 fonts (or two animation pseudo-fonts) may be used as the secondary font. These are set using the font group browser/editor. **FIXED**

MA 1 – 30 (font Motion program A) This is a special register paired with the pseudo-font ANA. If ANA has an attribute of 1 or 2 for primary or secondary font), then register MA is used to select one of a number of "canned" animation (motion) sequences involving two or more of the animation fonts. These are explained in the next section. **FIXED**

MB 1 – 30 (font Motion program B) This is a special register paired with the pseudo-font ANB. If ANB has an attribute of 1 or 2, then register MB is used to select one of a number of "canned" animation sequences involving two or more of the animation fonts. **FIXED**

RA 1 – 10 (animation Rate, program A) This register is used to select one of ten speeds to play the animation sequence selected with register MA and pseudo-font ANA. The speeds are in approximately 20% steps. **FIXED**

RB 1 – 10 (animation Rate, program B) This register is used to select one of ten speeds to play the animation sequence selected with registers MB and pseudo-font ANB. The speeds are in approximately 20% steps. **FIXED**

OP 0 – 2 (OPT button use) This register selects which of four operational parameters are controlled by the OPT button when not in the menus. For values 1-3 this allows a more direct way to browse or experiment with settings compared to changing the registers or font by means of the menu structure. These are the values for this register and the usage assigned to the OPT button:

0 OPT button toggles blanking mode on and off (see BA and BZ registers or BA and BZ serial commands)
1 OPT button directly alters register RA or RB animation speed if animation is active
2 OPT button directly alters register MA or MB animation sequence selector (1 to 25)
3 OPT button directly selects (changes) the primary or secondary font, in the order of the Font list.

Note that the changes affect the current display context. If, for example, the secondary font is being displayed, and OP = 3, then the secondary font will be changed. If the display time for secondary font expires and the display returns to primary font, then the OPT button will change the primary font. In other words, the OPT button affects the current context if applicable. Note that OPT=3 font changes are temporary, and will not be saved to the font menu or to nonvolatile RAM for use after power outages. **FIXED**

BA 0 – 23 (Blanking-blinking start) This register holds the hour when the display will be turned into the low-power tube-saving mode. When this mode is valid, the display will be totally blank, blank except for two brief flashes once every ten seconds, or dimmed. See register BF for the type of blanking. **FIXED**

BZ 0 – 23 (Blanking-blinking end) This register holds the hour when the display will resume its normal visible operation. If blanking is not desired, set both BA and BZ to the same value, which reduces the duration of blanking to zero hours. **FIXED**

BF 0 – 2 (Blanking Format This register controls the display mode when blanking has been activated by the BA register setting. A value of zero sets the display to wink the time in two short flashes every ten seconds. A value of 1 makes the display totally off when blanked, and a value of 2 selects the dimming mode for blanking; the display will operate normally but appear very dim. These modes are for conserving the lifetime of the nixie tubes, which regardless, can be expected to be very long, up to five to ten years, possibly even longer.

SU 0 – 2 (Seconds Underline) This register controls whether the underline segment in the nixie tubes will show a progress bar for seconds, or remain blank. If the value is zero, the underline seconds will not be used. A value of one allows stepping every 15 seconds (the first second is off, then one bar is added each 15 seconds). If the register has a value of two then the bar appears more linear, adding a segment every 12 seconds to form a 5-step sequence (none, one, two, three and four bars). **FIXED**

DP 0 – 9 (Date Program) This setting determines if and how often the date will appear in place of the time on the display. . **FIXED**

0 no date time display used (default)
1 on for 1 second every 20 seconds
2 on for 2 seconds every 20 seconds
3 on for 1 second every 14 seconds
4 on for 2 seconds every 14 seconds
5 on for 1 second every 9 seconds
6 on for 2 seconds every 9 seconds
7 on for 1 second every 6 seconds
8 on for 2 seconds every 6 seconds
9 on for 3 seconds every 6 seconds (these intervals, 6, 9, 14, and 20, are in 49% steps)

DC 0 – 4 (Date Characteristic) Selects among several alternate display characteristics to help identify the presence of the date display as opposed to the normal time display. Note that the font selected with an attribute of 3 can also be used to differentiate the date display, and it is added to the identifier selection in this register; if no font is selected for date, the current primary font is used. Animated fonts ANA and ANB are not available for the date display. **FIXED**

The characteristic display options are:

0 no display modifications (default)
1 winking four times per second
2 spastic display (makes it appear that there is a defect in the electronics)
3 winking and spastic modes combined

DF 0 – 1 (Date Format) A value of zero selects the format MMDD and a value of 1 selects DDMM. **FIXED**

ZH 0 – 23 (alternate time display, Hours) The Z is taken from "Zulu" time, which is an historic abbreviation for what is now UTC (or GMT). However, this feature is not only limited to UTC. Any alternate time zone may be used including those offset by 15, 30 or 45 minutes (the offset can be in any integer minutes in case you want to track the sun time, based on "local high noon"). There is no calendar association, so the offset is always positive, with 18 being the setting to achieve –6 hours, for example. Also, it remains fixed through daylight savings time changes (i.e., it does not follow the change); no separate rules are provided for such a change; if the alternate time zone desired uses daylight savings time, it will have to be manually changed twice per year. **FIXED**

ZM 0 – 59 (alternate time display, Minutes) This sets the minutes offset from the main time display. As noted above, the offset is always positive, so if the difference is negative, you must subtract it from 24 to get the value for registers ZH and ZM. For example, if you lived in Darwin, NT, Australia, which is UTC+9:30 and you wanted to have the alternate time display for Nepal (UTC+5:45) the difference is local time –3:45. To achieve this unusual offset ZH is set to 20 and ZM is set to 15. **FIXED**

ZP 0 – 9 (alternate time display Program number) This selects one of nine preset programs for the alternate time display. If set to zero there will be no alternate time display and all other Z-registers will be ignored. **FIXED**

The preset program numbers and their timings are:

0 no alternate time display used (default)
1 on for 1 second every 20 seconds
2 on for 2 seconds every 20 seconds
3 on for 1 second every 14 seconds
4 on for 2 seconds every 14 seconds
5 on for 1 second every 9 seconds
6 on for 2 seconds every 9 seconds
7 on for 1 second every 6 seconds
8 on for 2 seconds every 6 seconds
9 on for 3 seconds every 6 seconds (these intervals, 6, 9, 14, and 20, are in 49% steps)

ZC 0 – 4 (alternate time Characteristic) This register selects among several alternate time display characteristics to help identify the presence of the alternate time zone display as opposed to the normal time display. Note that the font selected with an attribute of 4 can also be used to differentiate the alternate time display, and this selection is in addition to the identifier selection in this register; if no font is selected

for the alternate time display, the current primary font is used. Animated fonts ANA and ANB are not available for the alternate time display. **FIXED**

The characteristic display options are:

0 no display modifications (default)
1 winking four times per second
2 spastic display (same effect as with the date display)
3 winking and spastic modes combined.

FR 1 – 4 (line/mains FRequency selector) This register unequivocally sets the clock to match the power line (mains) frequency in use. The initial default value is zero, which selects 30 Hz. (This means the first time the clock is run, if register FR is not set, the time will run very fast, which is an alert that this register must be set. If FR is not set the clock will display FRXX instead of OK at the end of its Hello message.) The four possible values that can be stored into FR are:

1 50 Hz power from use of an AC "wall wart" power supply.
2 60 Hz power from use of an AC "wall wart" power supply.
3 100 Hz for use with a DC "wall wart" power supply run on 50 Hz mains.
4 120 Hz for use with a DC "wall wart" power supply run on 60 Hz mains.

Note that the last two selections require special changes to the circuit that are not standard.

CU 0 – 2 (Clock Update mode) This determines the fundamental operating mode of the GeekKlok with respect to how and where accurate time is counted, maintained and updated. If this register is set to zero, the RTC chip is assigned the role of being the master timekeeper. This would be done if the 50/60 Hz power line is unreliable for timekeeping, such as when running from a generator or in a location where the power line frequency is unstable (e.g., on as yacht). The GeekKlok follows the accuracy of the crystal oscillator (Y94). If this register is set to 1, the normal default value, then the 50/60 Hz power line is the standard for timekeeping, and the RTC is updated from the power line time counters once each 24 hours (at 03:30). If the CU register is set to 2 the source of timekeeping is external, as supplied by a 1 Hz pulse (from a GPS module, for example) connected to the GeekKlok. The RTC is updated once each day, also at 03:30. Use of this external reference requires adding wiring and making additional PCB modifications. **FIXED**

LZ 0 – 2 (Leading Zero) This register controls whether the leading zero (tens hours) is blanked or shows when the digit is zero. If the value is zero, the leading zero is always blanked. If the value is 1 then it is blanked for static fonts, but is not blanked for animated fonts. If the value is 2 then the leading zero is always shown (unblanked). This is only active in 12-hour display mode. In 24-hour display mode the leading zero is always shown. Many animation sequences are most interesting if all four characters are used in the animation. **FIXED**

The next ten registers control the automatic, rule-based daylight savings time (summer time) correction system.

AD 1 – 7 (Autumnal Day) This register holds the day of the week for the autumnal revocation of daylight savings time. The day convention must agree with register DW, and each day begins at midnight. **FIXED**

AM 1 – 12 (Autumnal Month) This register sets the month when the change back to standard time occurs. **FIXED**

AW 1 – 5 (Autumnal Week) This register determines the week, within the month specified in register AM, when the change to standard time occurs. If the value is 1 and register AD indicates Sunday, then the change will occur on the first Sunday of the month specified by register AM and at the hour specified by register AH. The value of 4 indicates the fourth week and a value of 5 represents the last week in the month when AD occurs, which can actually be the fourth or fifth week, depending on the date and year. Continuing the example from above, a value of 2 for AW means the second Sunday of the month, 3 is for the third Sunday of the month, 4 is for the fourth Sunday of the month and 5 is for the last Sunday of the month. **FIXED**

AH 0 – 23 (Autumnal Hour) This register sets the hour when the change to standard time occurs. If this register is set to 2, then at 02:00 on the day, week and month specified by AD, AW and AM the time will change to 01:00. One hour later it will not change again, since register ST is the determining condition as to whether daylight savings time is in effect or not. Register ST is changed by the clock when the automatic adjustment occurs. **FIXED**

VD 1 – 7 (Vernal Week) See discussion for AD above. **FIXED**
VM 1 – 12 (Vernal Month) See discussion for AM above. **FIXED**
VW 1 – 5 (Vernal Week) See discussion for AW above. **FIXED**
VH 0 - 23 (Vernal Hour) See discussion for AH above. If this register is set to 2, then at 02:00 on the day, week and month specified by AD, AW and AM the time will change to 03:00. **FIXED**

SV 0 – 1 (Savings time Valid) This register determines if daylight savings time (summer time) is valid for your location. A value of zero means that none of the registers for auto adjustment described above will be acted upon to attempt automatic correction, regardless of their contents. If daylight savings time is legally mandated in your area, set this register to a value of 1. **FIXED**

ST 0 – 1 (Savings Time) This register must be set to properly indicate whether the present time and date being set is standard time or daylight savings (summer) time. This register is changed by the clock when automatic correction takes place. If the value is not correct, the automatic correction will be skipped

the first time the correct day, month and hour arrives, but will be correct every time thereafter. A value of zero indicates winter time and a value of one indicates summer time (daylight savings time). **VARIES**

YY 0 – 99 (current Year) This register is for the current year (last two digits). **VARIES**

MM 1 – 12 (current Month) This register is for the current month. **VARIES**

DD 1 – 31 (current Date) This register sets the date of the month. The browser/editor does not check the month to select the maximum value. If you set the date to 31 for a month with 30 or fewer days, it will revert to the last valid day of the month. **VARIES**

DW 1 – 7 (current Day of Week) This register sets the day of the week. You can use 1 for Sunday and 7 for Saturday, which we suggest, but any other day-one can be used. The convention used must be uniform for the automatic daylight savings time (summer time) correction registers (AD and VD). Date settings can be ignored if the display of date and summer time correction is not required, but if either date display or autumn-spring summer time correction is requiredm then the date must be correctly set. **VARIES**

FONT SETTINGS

Fifty different fonts are available for selection, to be displayed as a primary or secondary font for time, and to help differentiate the display of date and month and an optional alternate (Zulu) time zone, if desired.

To enter the font setting editor, press SET once (to display the **TIME** prompt), a second time to display **REGS** and then a third time. The prompt will show **FONT** on the display. Pressing ADV will then enter the font setting editor.

The font editor starts at the beginning of the list of 50 fonts (see Font Catalog below). The SET button advances from font to font in the order shown in the FONT CATALOG. Holding the OPT button down reverses the direction of stepping with the SET button, allowing you to move forward and backwards through the list.

The ADV button will advance the font usage number, or attribute, cycling to 1, 2, 3 or 4, then back to 1. Only one font can have each of these attributes. For example, if you set NOR to 1 and then set MIA to 1, NOR will automatically return to zero.

Attribute 0. The font is not used. Fonts with an attribute of zero can be changed to 1–4, but not back to zero.
Attribute 1. The font is the primary font and will display for a duration specified by the **FP** register (x10 seconds).
Attribute 2. The font is the secondary font and will display for a duration specified by the **FS** register.
Attribute 3. The font is used for displaying the date and month. Refer to the description of registers **DP** and **DF**.
Attribute 4. The font is used for displaying the alternate "Zulu" time zone data. See registers **ZP**, **ZC**, etc. This font is also used to display the time when blanking of the display is active (see register BA, BZ, OP and serial commands BA and BZ).

Fonts can be activated to have an attribute of 1 through 4, but can not be deactivated except through activation of a different font for the same usage.

At the end of the list of 50 fonts there are two additional "pseudo-fonts" shown in the editor/browser. These are named ANA and ANB (anamation A and animation B), and are paired with user registers MA and MB, respectively. ANA or ANB can only be assigned to primary and secondary time displays; due to the brevity of the date and alternate "zulu" time displays, ANA and ANB can not be assigned an attribute of 3 or 4. The actual animation sequence that will play is then set by register MA and MB. It is with the actual font sequence number in MA or MB that the true fonts are called for use. The user has no ability to program his/her own variations of animation sequences; the "canned" sets already have quite a number of variations. The fonts that are used in the animation sequences include the four windblown "Burn" fonts (BNW, BNE, BSE, BSW), designed by Dr. Peter Csaszar, and Normal, Missing in Action, Atrophy, Moocher, Visitor, Intruder, Warped, Motley Inverted, Reversed. These are arranged in 25 different sequences that are visually interesting.

For more immediate access to ANA and ANB, press the OPT button while using SET to access the fonts in the reverse order, going fromNOR to ANB and ANA.

See the Font Catalog section for a graphical description of all fifty fonts.

During display of primary or secondary fonts, the OPT button can be used to browse forward through all of the fonts, if register OP has been set to a value of 3. This makes it easier to browse the font list without having to refer to the fonts by abbreviated name and/or having the FONT CATALOG handy for reference. The OPT button browse includes animation fonts ANA and ANB. OPT button browsing does not change the FONT register settings; they are only temporary, meant for sampling.

CLOCK ACCURACY

In most locations the timebase derived from the mains power line, 50 or 60 Hz, is very accurate over a long-term basis. The accuracy may drift during portions of any 24-hour period. For example, the frequency may be lower during the peak load hours, and then speed up to compensate during the low-load early morning hours. While the GeekKlok has an internal Real-Time Clock (RTC) chip that runs from a battery, it is typically not anywhere as accurate, over a long period, as the time from the 50/60 Hz line frequency. For this reason the RTC chip is updated once daily with the correct time as counted from the 50/60 Hz power line. See the description of the register CU setting.

If the power fails, the time based on the power line is lost. When power is restored again, the internal time is taken from the RTC. This is why the RTC is updated daily, since if it were not updated its time would drift. From investigations of power line accuracy in North America and Europe, accuracy over a 24-hour period is excellent, and updating the RTC every 24 hours will result in very accurate backup time data in the event of a power failure and restart cycle.

CIRCUIT DESCRIPTION

The heart of this device is the microcontroller, a Motorola MC68HC705C8A device that has 7684 bytes of program storage and 176 bytes of RAM. The program is stored in one-time programmable memory and is protected from copying or downloading. If you attempt to read the contents of the chip you will fail and might damage the part. The program is copyright by the author and represents an investment of considerable effort.

This clock uses multiplexing to display characters on the nixie tubes. Only one tube at a time is turned on, but the rotation or sequencing occurs so quickly that you see a steady display of four characters. Multiplexing requires 15 driver transistors to turn on each segment (cathode), and four transistors to connect each tube's anode to the high-voltage power in sequence. Cathodes and anodes that are turned off are brought to the potential of about 85 volts through "pull-middle" resistors. This is done to eliminate ghosting effects, where a tube may show a different character or part of a character dimly besides the bright segments that show the desired character. The resistors discharge residual charges of the parasitic capacitance formed by the tube's internal form and the tracks on the PCB. That these resistors also help reduce acoustical noise or "singing" that can occur in multiplexed nixie tube displays, due to electrostatic charge forces exerted on the parts of the tubes..

The time counting is based on the 50 or 60 Hz power line (unless disabled). The AC is filtered and sent to U93 and associated components. This is a 50 or 60 Hz oscillator that is phase-locked to the mains power supply. If for any reason the AC drops out for a brief moment (not long enough to shut down or reset the microcontroller), the oscillator will free run at the line frequency, thus continuing supply timing pulses. The AC might drop out when a heavy motor (like an air conditioner) starts up or when a distant lightning strike disturbs the power line voltage for a period ranging from one tenth of a second to over one second. Without the "flywheel" effect of this oscillator, each such brief dropout would cause loss of pulses and consequently a general slowdown of the clock. This circuit also eliminates severe noise pulses from disturbing the timing. The output of U93 is sent to the timer capture input of the microcontroller.

The AC power is rectified and supplied to a linear regulator (U1) to generate 5 volts for the logic. The unregulated power is used by the switchmode power supply, employing a MAX771 IC operating at about 300 kHz to boost the 12 volts (approximately) to about +170 to 180 volts to power the nixies.

Another very important IC is the RTC (Real Time Clock), U94, which operates from a battery when power is removed from the GeekKlok. Under battery operation a 32768 Hz oscillator operates to keep time and date (calendar). When the GeekKlok is powered up the date and time, plus all of the user settings in registers and fonts, is recovered from U94. This IC is not generally used for keeping time information because the accuracy of the 32768 Hz crystal is several orders of magnitude worse than the average of the AC power line, provided you are operating the unit in a country where such power is kept accurate and stable by the power generating authority. See the section on CLOCK ACCURACY for more information.

POWER CONNECTION and POWER CONSUMPTION

The GeekKlok requires 12 volts AC for power, and consumes 350 mA maximum, which is 4.2 watts. The unit does not generate significant heat and canbe operated in most inside locations without concern for overheating. Power must be supplied by a step-down power transformer pack approved for use in your country. If the display is blanked the power consumed drops to below 1 watt. The internal battery is expected to last for at least 7-10 years. The nixie tubes normally will last for from eight to 20 years. They are operated at well below their maximum ratings for the longest possible lifetime. The GeekKlok designer, Raymond Weisling, has used these in products since 1973 and is well familiar with their characteristics and proper usage.

FONT CATALOG

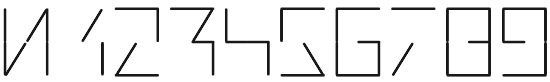
The first sixteen fonts, shown on this page, are available in both static forms and included in preset animation sets (font names ANA and ANB), which will be described at the end of the font catalog section. Each font name is shown with its three-character abbreviation that is used in the font editor/browser.

The first group starts with the "native" number set, **Normal**, followed by variations on this which subtract or add segments.

Missing in Action and **Atrophy** have segments removed. **Visitor**, **Moocher** and **Intruders** have added segments (adding one, one and two segments, respectively).

Normal Numerals NOR

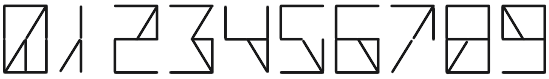
Missing in Action MIA



Atrophy ATR

Visitor VIS

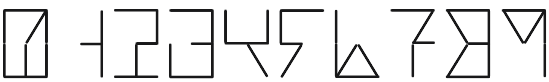
Moocher MOO



Intruders INT

The next set use mirror reflection, or involve changes in shape from the normal font, usually with one to three segments affecting the shape change. These four fonts, and the above six, occur mixed in different animation sequences to achieve different effects.

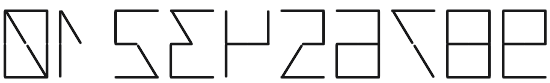
Warped WAR



Motley MOT

Inverted INV

Reversed REV

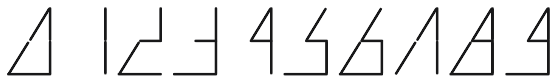


The following four fonts were designed as a unified animation group. They appear in different animation sequences, but never mixed with the first ten in the left column. They can, of course, be selected to be static fonts as well.

These are clever designs by Dr. Peter Csaszar, specifically for the GeekKlok. The BURN designation applies to the corner that is "burned out" and where no segments can illuminate. They can also be called "Windblown" characters because they are tilted as if blown by a gale. Peter's fonts actually inspired the development of the animation feature that is unique to GeekKlok.

BurnNE BNE

BurnNW BNW



BurnSW BSW

BurnSE BSE

The following 34 fonts are only available as static display fonts, though with primary and secondary font selection and use of the serial port Jump-to commands, they may to a limited degree be animated. **Normal**, shown above, is the design used in our Four Letter Word sculpture. It is slightly different from the original Burroughs character set for the B-7971 Nixie Tube, which is repeated here as "**Burros**". A standard seven-segment font, stark and ugly compared to those that use the diagonal segments, is also included. The font **Digits** starts to deviate from the classical numerals. More deviants are shown further below.

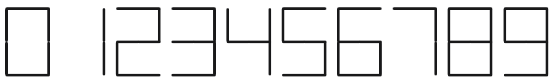
Dancer uses a kind of semaphoric scheme (inspired by, though not based on, real semaphore signalling codes). The upper arms form the coded part, while the lower "legs," if alternated, make it appear to be a (well, headless) stick figure dancing.

Dancer Left DNL

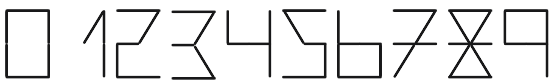
Dancer Right DNR

Burros BUR

Seven-Segment SEV



Digits DIG



The **Rotated** font uses basic numeral shapes that have been rotated anti-clockwise 90°, adjusted to fit the segment lengths. **Twister** combines rotation with reflection.

Rotated *ROT*

Twister *TWI*

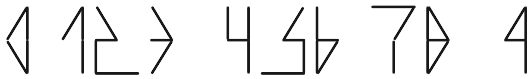
This group of characters is distorted by skewing, leaning or otherwise altering the shape of the basic numerals. They can still be read without "special training"

Deco *DEC*

Miring *MIR*

Skewed Up *SKE*

Tall & Thin *TAL*

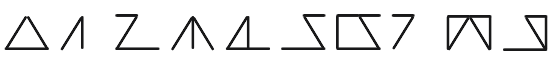


Distortion and reshaping continues. Here we have a set of tiny characters, either in normal orientation or rotated by 90°. **Hi-Lo** is a combination of **Low-Boy** and **Hi-Girl**. The clock could be set on end to read the rotated ones, but then they couldn't be mixed with others.

Rotated CW *RCW*

Tiny Rotated *TRO*

Low-Boy *LOB*



Hi-Girl *HIG*

Hi-Lo *HIL*

Pythagoras is based on two or three triangles, attempting to use shapes that mostly are recognisable as normal numerals or their basic outline shape (except 3). **Doodles** starts to show significant deviation from normal numerals, but still retains some basic shapes for the majority of digits.

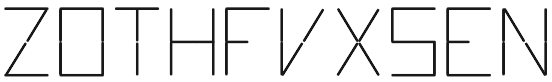
Pythagoras *PYT*

Doodles *DOO*

Language-Based Characters

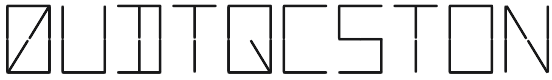
Turning now to words for numbers, and letters from those words, these characters provide hints for the numerals that they represent. **ASCII** is actually based on English, taking the first or other prominent letter as a substitute for the number (**Z**ero, **O**ne, **T**wo, t**H**ree, **F**our, fi**V**e, si**X**, **S**even, **E**ight, **N**ine). The same scheme is approximately true for **Deutsch** and **España**, though not totally accurate. The other letter-hinting fonts are based on English but use other parts of the words to provide hints for the number.

ASCII *ASC*



Deutsch *DEU*

España *ESP*



Suggestion *SUG*

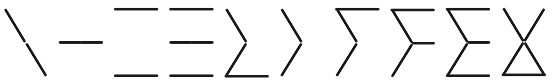
Hints *HIN*

Alternate Script Systems

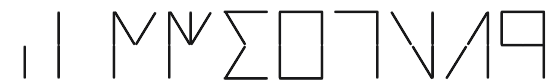
In this set we have two version of Roman numerals (standing and lying down), plus true **Arabic** numerals (as used in modern Arabic writing), and a set suggesting Chinese or Japanese written numerals (close, no cigar?). The **Klingon** set is an approximation of the canonical Klingon language taken from Star Trek.

Roman *ROM*

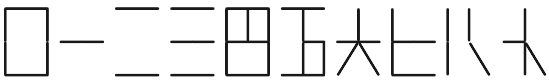
Nero *NER*



Arabic *ARA*



Tokyo TOK

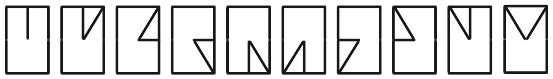


Klingon KLI

Morsee (Morse-See) is a visual representation of Morse Code, read from left to right and top to bottom, with vertical lines for dits and horizontal lines for dahs. **Watchout** suggests the hands on a watch, but what is important is the position of the triangle. **Wedges** uses a similar process of clockwise triangle development for digits. **Fingers** represents one to five fingers, then for six to nine the left hand is brought in, palm down, to represent five, with the fingers counting again. **Starburst** adds segments to build only radial lines from the centre. **Binary** uses the four quadrants of the display to represent binary 8-4-2-1 weighting, moving in an anti-clockwise manner, the way that the quadrants are numbered in a two-dimensional Cartesian plane. With **Pie** the triangular portions removed are similar to the triangles in Wedges, but the order is reversed. And finally **Skarnox** is an invented alien font that uses its own "logical" system to represent the numbers. Most of these fonts are "private" fonts that the GeekKlok owner will have to memorize as a challenge, primarily to impress skeptical friends.

Morsee MOR

Watchout WAT

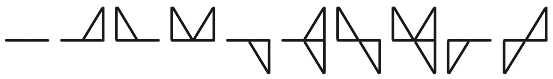


Wedges WED

Fingers FIN

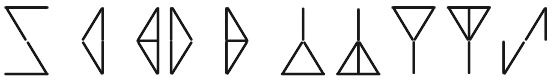
Starburst STA

Binary BIN



Pie PIE

Skarnox SKA



Selecting Animations

At the end of the font list in the editor/browser there are two pseudo-fonts named **ANA** and **ANB**. These are selected for display just like any other font, but they actually represent a collection of fonts in a preset timed sequence that will play when the pseudo-font is enabled for appearance. These fonts can only be set to primary or secondary (1 and 2), and are unavailable for date and zulu time display modes.

Animation Font A *ANA*

Animation Font B *ANB*

The font editor/browser setting is for primary or secondary. ANA can be secondary and ANB can be primary; the A and B designation is just to differentiate them as separate entities.

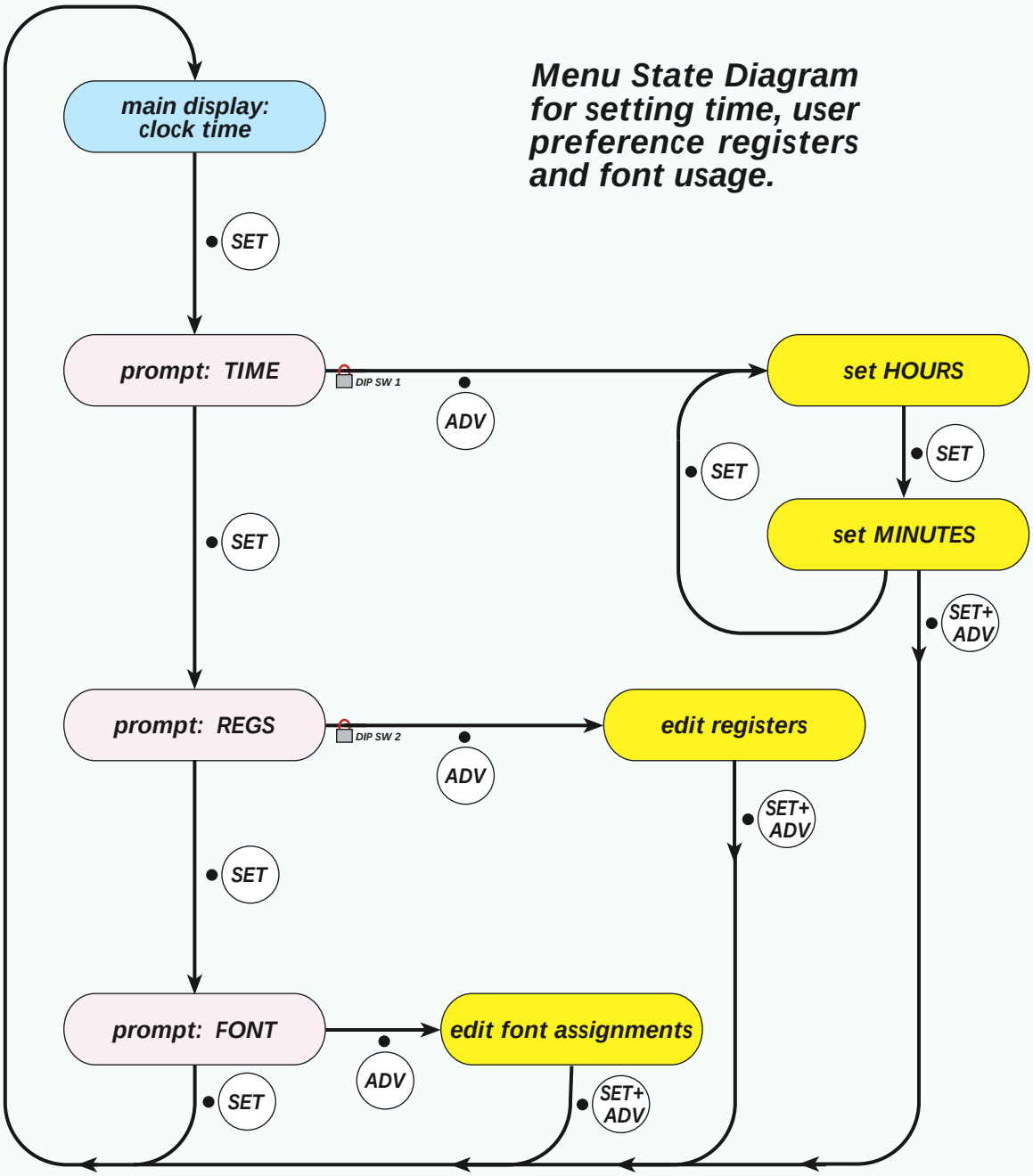
In order to actually select which fonts are used in the animation, registers have been paired with ANA and ANB for selection of the animation program or sequence, and additional registers have been paired for selecting the speed.

Consequently the pseudo-fonts are paired with these registers:

font	program	speed
ANA	***MA***	***RA***
ANB	***MB***	***RB***

The list below describes the animation programs available. The number of the sequence is the value that must be stored into the **MA** or **MB** register. The speed or rate desired is stored in the **RA** or **RB** register. L-R means left to right, L-R-L-R means movement in both directions, left to right and then reversing, right to left.

1 Four-step rotary motion of all four windblown (burn) fonts.
2 BNE & BNW flip-flop of all four characters.
3 BSE & BSW flip-flop of all four characters.
4 BNE & BNW opposing motion in hours and minutes.
5 BSE & BSW opposing motion in hours and minutes.
6 BNE & BNW flip-flop alternating hours and minutes.
7 BSE & BSW flip-flop alternating hours and minutes.
8 Left-to-right ripple rotation of all four windblown (burn) fonts.
9 BSW & BNE left-right-left-right (L-R-L-R) ripple.
10 BNE & BNW in L-R-L-R ripple with pause at each end.
11 BSE & BSW in L-R-L-R ripple with pause at each end.
12 BSE & BSW in slow L-R-L-R ripple, fast return at ends.
13 BNE & BNW in slow L-R-L-R ripple, fast return at ends.
14 BNE & BSE in slow L-R-L-R ripple, fast return at ends.
15 BNE & BNW in L-R ripple-kick motion.
16 BSE & BSW in R-L ripple-kick motion.
17 All four windblown fonts in L-R and R-L ripple-kick motion.
18 All four windblown fonts in L-R-L-R ripple, random durations.
19 NOR, MIA and ATR fonts in L-R-L-R ripple, long NOR periods.
20 NOR, MOT and VIS fonts in L-R-L-R ripple, long NOR periods.
21 NOR, MIA, ATR, VIS, MOO, MAR, MOT randomly appearing.
22 NOR, WAR and MOT randomly timed, HH & MM pair changes.
23 NOR, INV and REV in L-R-L-R ripple.
24 NOR (long), occasionally all change to MIA, ATR, WAR and MOT.
25 MIA, ATR, VIS, WAR, MOT but no NOR, random, with spastic effect added. (This is the "major defect simulation" mode.)



Key to Symbols:

= Editing of numeric values with ADV button: 1x click forward, reverse if OPT button held down. OPT + SET button reverses order of register or font access. Hold button down for autostep.

SET = 1x click of SET button chooses this path.

ADV = 1x click of ADV button chooses this path.

SET+ ADV = 1x click of SET and ADV at same time button chooses this exit path.

= path locked out if specified DIP switch set to ON.