ED 346 665 EC 301 277

AUTHOR	Kelso, David; And Others
TITLE	Alternate Access Interface for Mouse and Touchpad Input. Final Report.
INSTITUTION	Wisconsin Univ., Madison. Trace Center.
SPONS AGENCY	Department of Education, Washington, DC.
PUB DATE	89
CONTRACT	G008730317
NOTE	71p.
PUB TYPE	Reports - Descriptive (141) -- Reports - Evaluative/Feasibility (142)
EDRS PRICE	MF01/PC03 Plus Postage.
DESCRIPTORS	*Accessibility (for Disabled); Computer Uses in Education; Elementary Secondary Education; *Input Output Devices; Keyboarding (Data Entry); Mainstreaming; Microcomputers; *Physical Disabilities; Technological Advancement
IDENTIFIERS	*Trace Transparent Access Module

ABSTRACT

This final report describes a federally funded project (September 1987-August 1989) which developed the "Trace Transparent Access Module," a device which interfaces with standard computers thus allowing children with physical disabilities to participate fully with nondisabled children in classrooms. The module is a small, microprocessor driven device which emulates the activity of the keyboard and mouse on several models of Apple and IBM computers. It enables individuals with disbilities who cannot use a standard keyboard and mouse to input keystrokes and mouse movements, thus allowing access to all the same software programs used by their peers. In addition to the module, a General Input Device Emulating Interface (GIDEI) standard was developed which enables manufacturers of electronic communication aids to use RS-232 serial commands to control the operation of the keyboard and mouse. Field testing with one disabled individual and two clinicians has demonstrated the usability of the user manual, and the functionality and reliability of the device. The bulk of the report consists of attachments including the grant proposal, photographs of the device, the device manual and schematics. (DB)

FINAL REPORT

Alternate Access Interface for Mouse and Touchpad Input

Grant No. G008730317, U.S. Dept. of Education

Copyright © 1989
Trace Research and Development Center
Waisman Center
University of Wisconsin-Madison
1500 Highland Avenue
Madison, WI 53705

Alternate Access Interface for Mouse and Touchpad Input Project September 1987 - August 1989

Project Team
David Kelso
Charles Lee
Mark Novak
Joseph Schauer
Gregg Vanderheiden

Trace Research and Development Center
Waisman Center
University of Wisconsin-Madison
1500 Highland Avenue
Madison, WI 53705

Funding for this project was provided by Grant No. G008730317, U.S. Dept. of Education

Abstract

As the use of computers continues to become an integral part of the educational environment, the need for all children to be able to use a computer in a school program is of great importance if those children are to have an equal opportunity for quality education. However, there are many physically handicapped children who are unable to use a computer because they cannot use the standard keyboard or mouse. As a result of this inaccessibility of computers, these children are unable to participate equally with non-handicapped children in computer assisted learning activities, and the benefits of mainstreaming these children cannot be fully realized. In addition, the use of a computer to augment communication and writing for the physically disabled child is limited.

The "Alternate Access Interface for Mouse and Touchpad Input" project, Grant No. G008730317, was funded through an award under Technology, Educational Media, and Materials for the Handicapped program of the U.S. Department of Education. This program calls for "innovative adaptations" of technology to increase the "accessibility to educational opportunities" for individuals with physical, sensory or cognitive disabilities.

The goal of this project was to apply technology in the development of a solution that would enable children with physical disabilities to participate fully with non-disabled children in classrooms. The result was the development of the *Trace Transparent Access Module*, a device which interfaces to standard computers, without modifications, to allow both disabled and non-disabled children equal access to the computer.

The Trace Transparent Access Module (T-TAM) is a small, microprocessor driven device which emulates the activity of the keyboard and mouse on several models of Apple and IBM computers. It enables the disabled individual who cannot use the standard keyboard and mouse to use a special electronic aid to input keystrokes and mouse movements into the computer, thus providing access to all the same software programs used by their non-disabled peers.

The project also included the evaluation and field testing of the device, along with a plan for distribution and marketing. The T-TAM prototype has undergone preliminary field testing and evaluation by several individuals and manufacturing representatives. Two major manufacturers of augmentative communication aids for people with disabilities have expressed the desire to manufacture the T-TAM and the Trace Center is continuing to work closely with them.

At this time, additional features and final modifications are being completed with the aid of new funding sources. Completion of the project is scheduled for January, 1990 with commercial production beginning in March.

Grant Requirements

In response to a request for proposals for grants under Technology, Educational Media, and Materials for the Handicapped program, administered by the U.S Department of Education, Office of Special Education and Rehabilitative Services, the Trace Center submitted a proposal titled, "Alternate Access Interface for Mouse and Touchpad Input" and was awarded Grant No. G008730317. The Compensatory Technology Applications (84.086P) RFP called for development of "innovative adaptations of hardware and software technology and the field-test evaluation of those innovative adaptations" which would "compensate for physical, sensory, or cognitive learning impediments in order to: (a) Alleviate the need to modify instructional materials and/or (b) increase the overall accessibility to educational opportunities for handicapped learners."

The RFP required the development of prototypes which demonstrated the use of technology in addressing the needs given above. Upon completion of development, evaluation and field testing was to occur along with "a plan for national marketing and distribution including a rationale supporting the modifications based on the field-test results."

The Trace Center proposal addressed this need through the development of an electronic interface which would allow severely physically disabled children to access standard computers used in schools. The interface would allow these children to participate fully in educational environments where computers are being used, and increase their opportunities for achievement.

Description of Problem Area

The Trace Center project identified these specific needs: (1) the need for standard input device (keyboard, mouse, etc.) emulators for computers commonly used in schools and other educational environments to make them accessible by severely physically disabled children, and (2) the need for modifications to standard input devices for computers commonly used in schools and other educational environments to make them accessible by moderately physically disabled children.

The use of computers in education and employment areas has been and continues to be on the rise. Computers have already become an integral part of the educational environment, with literally thousands of educational software programs available for teachers and students. The need for accessible school computers is of great importance if all children are to have an equal opportunity for a quality education.

Students with disabilities also need to be able to use the computer as a training tool. Many special education software programs are available and written specifically to help students with learning disabilities or students who as a result of physical disabilities have not previously had the opportunity to obtain basic educational skills. These programs are useless for the disabled student who cannot operate the computer.

One reason many physically disabled individuals cannot use the computer is because they cannot operate the standard input devices (for example, the keyboard and mouse) that come with the computer from the manufacturer. To complicate matters, many advances designed to ease the use of the computer and to make it more user friendly for able bodied users have created new barriers for people with physical disabilities.

Many types of electronic aids are available that are specially interfaced to the abilities of the severely physically handicapped child and which enable the child to communicate with others or have environmental control capabilities. And with the advent of the Keyboard Emulating Interface (KEI), these individuals have also been able to use their electronic aid for accessing computers. (The KEI is a device that can be thought of as a keyboard with no keys. It is connected to the computer just like a keyboard, and also to the user's communication aid. Keys are selected by sending encoded signals to the KEI from the aid. The KEI converts the signals into keystrokes which are then sent to the computer.)

Although these KEIs are available widely, they exist for only a limited number of computers (IBM PC, Apple II+, Apple IIe). The reason for this limited number is that each computer model requires a different KEI because of electronic differences in the way the keyboards work. Since a large amount of research and development time is required to create a KEI, and new models of computers have been coming out so fast, rehabilitation engineers have been unable to keep up.

Another major problem is that the KEIs still do not address the need for access to newer input devices such as the mouse and the touchpad or touchscreen. The Apple II family of computers is used extensively in school systems, with more and more Apple IIGS computers being purchased. The Apple IIGS uses a mouse. The Macintosh family of computers and IBM Personal System/2 series also use a mouse as a standard input device and therefore are inaccessible to severely physically disabled children. These computers are quickly becoming integrated in elementary schools, high schools and universities.

For the moderately physically handicapped individual, operation of the keyboard may be possible, but limited. Often times requirements by the computer operating system or application software pose barriers: for example, a person with one arm may find it impossible to activate multiple key combinations frequently used to restart the computer.

In some cases, solutions already exist for enabling the moderately handicapped child to access some computer software. Keyguards, mechanical latches, and special software modifications to operating systems have been used. However, these solutions do not always work for all computer models and operating systems. There are still many handicapped children who are unable to use the computers even with these modifications.

The primary focus of this grant project was to develop a solution that would remove present barriers to computer accessibility for the severely physically disabled individual. As the project progressed, additional funding was obtained from outside sources which enabled us to address a much wider scope that included moderately physically disabled individuals as well.

Objectives

The Trace Center project had three primary objectives:

1) To provide an accessible Apple II family of computers through the development of a General Input Device Emulating Interface (GIDEI) for the Apple IIGS computer.

2) To create a GIDEI standard.

3) To provide an accessible Apple Macintosh and IBM PS/2 line of computers through the development of a GIDEI for these computers, if time permitted.

Finally, as the project progressed, a fourth objective was developed and completed under separate funding:

4) To provide additional accessibility to IBM PS/2 computers through the development of special keyboard enhancement functions within the GIDEI device.

The goal of this project was to create an accessibility device which would meet the needs identified above. However, due to the broadness of the needs, it was necessary to focus our attention on the Apple IIGS computer first. This computer was selected since the Apple II is the most widely used family of computers in the elementary and high school settings. Very many programs have already been written for these computers and the Apple IIGS is the model being used to develop the next generation of special education and general education software.

Another factor in selecting this computer was that it was the first Apple II computer to incorporate the mouse as a standard input device. Programs were already available that required the use of the mouse and more would be developed.

Development of an access device for the Apple IIGS will open new opportunities for those children previously unable to use this computer at their school. Also, school districts concerned about equal opportunities for all will not be deterred from buying this newer and more powerful computer due to a concern that it would leave the physically disabled children behind. This should enhance the ability of the educational program to make use of improving technology.

Another goal of the project was to develop a standard set of interface specifications for input device emulators. These specifications are called the General Input Device Emulating Interface Standard. This standard assists those involved in creating input emulating devices for other computers to assure that they are compatible with each other. The development and adoption of a standard is important for manufacturers of electronic aids so they can proceed with the confidence that their design of an aid will work with all GIDEI devices. This benefits the aid manufacturer by expanding the marketable features of their aids, and encourages the development and support of input emulators for other computers.

A third objective of the project was to incorporate into the device the capability to emulate standard input devices for the Macintosh and IBM PS/2 families of computers. Although initially these computers were not widely used in elementary and high schools, we did not want schools to be deterred from buying newer computers because they were inaccessible to the physically disabled students. Similarly, disabled students should not be limited as to the types of computers they can use.

The fourth objective (developing enhancements to the function of the IBM PS/2 keyboard) was included after it became clear that these features could be added without design changes and would benefit many more individuals. The new efforts would allow moderately physically disabled children to invoke special features that changed the operating characteristics of the keyboard. These features would allow these children to use the same keyboard used by non-disabled children to operate programs on IBM PS/2 computers.

Summary of Project Work

Initial project work involved collecting information to identify the need for a GIDEI. Surveys were sent out to several school administrations in Illinois and Wisconsin to identity the types of computers in use and to determine future plans for purchasing computers. Results of these surveys indicated that many school systems already had Apple IIGS computers and would be purchasing more of them. In addition, they indicated that they would also be purchasing Macintoshes and IBM (PS/2) computers.

Also during this period, discussions were held with manufacturers of electronic aids for people with disabilities. Information was collected on expressed needs for a GIDEI device, potential features the device would have, manufacturing and marketing capabilities of these manufacturers, and any potential limitations which could affect the success of a GIDEI product.

Results of these discussions indicated that there was a need for a hardware emulator for Apple IIGS, Macintosh and IBM computers. Since new operating systems were being developed, the software KEIs in existence were becoming obsolete. Manufacturers expressed concern that any new emulating interface should be compatible with existing devices and that design should take into account their manufacturing capabilities.

Considering the input from the manufacturers contacted, a list of design specifications were developed and preliminary designs made. Initially, separate designs were developed for Apple comput ,rs and IBM PS/2 computers. However, after considering functionality, cost and marketing concerns of the manufacturers, a single device was designed allowing connection to both the Apple and IBM systems. (Schematics and software listings for this system are included in attachments to this document.)

The device which was designed was given the name Trace Transparent Access Module (T-TAM). The T-TAM is able to emulate the mouse and keyboard on all Apple computers that use the Apple Desktop Bus (ADB) input device interface. This includes the Apple IIGS, Macintosh SE and SE/30, and all Macintosh II series computers. The T-TAM also emulates the keyboard and mouse on all IBM PS/2 computers. Additionally, it emulates the keyboard on IBM AT computers and compatibles.

The T-TAM is connected between the standard keyboard and mouse, and the computer. Non-disabled individuals are able to use the mouse and keyboard in a normal fashion, just as if the T-TAM were not connected at all. It was important that the T-TAM have this feature to gain acceptance in the schools and other places where computers are shared.

The severely physically disabled child uses the T-TAM by sending special commands to it through an RS-232 serial port. Typically, the commands are sent from a special electronic aid which the child uses for communication and environmental control. When the aid is connected to the T-TAM, the child is able to control the operation of the keyboard and mouse.

The T-TAM may also be used by moderately physically disabled children using the standard computer keyboard. Four important features have been programmed into the T-TAM which make this possible. These are 1) "sticky keys," which eliminates the need to physically hold more than one key down at the same time; 2) "filter keys," which filters out accidental key presses and allows for typematic rate adjustment; 3) "mouse keys," which allows you to operate the mouse pointer from the keyboard; and 4) "toggle keys," which gives audible feedback on the status of certain keys for people with impaired vision.

These features are activated by special key sequences which typical keyboard users would not perform. This is important in multi-user environments such as schools where many different types of users will need access to the same keyboard.

Developed in addition to the T-TAM was the GIDEI Standard. This Standard enables manufacturers of electronic communication aids to use RS-232 serial commands to control the operation of the keyboard and mouse. At this time, there is not a standard touchpad device, separate from the mouse, to emulate directly. However, the GIDEI has been designed to allow developers of special aids to simulate touchpad input.

The GIDEI standard was implemented in the T-TAM and is currently being assembled as a working document, one which is continually updated and revised as knowledgeable people in the field respond with comments and suggestions.

A manual was prepared to assist in connecting the T-TAM to an aid and to the target computer. It also describes the operation of the GIDEI portion of the T-TAM. Information is organized according to the capabilities of the user with separate sections for Beginning Users, Intermediate Users, and Advanced Users. Currently, a second version of the manual is being completed to better address the needs of the users who evaluated it and to incorporate the operation of the additional keyboard enhancement features.

Evaluation and Field Testing

Testing of the T-TAM was performed to evaluate different components within the T-TAM project.

One field test was developed to determine if users could learn how to setup and operate the T-TAM from the user manual without any expert assistance. The rationale behind this is that significant cost savings in product support could be demonstrated to prospective manufacturers if individuals required little or no support. This would result in lower costs for the device.

Three participants, one disabled individual and two able bodied clinicians, performed this testing procedure separately. Results of the test indicate that the manual by itself was insufficient in providing the user with enough information to connect and operate the T-TAM device. One of the subjects required technical assistance in connecting the T-TAM to their aid and the computer. Also, all subjects had difficulty understanding the concepts involved in operating the T-TAM and therefore they either did not know what was required of them in order to program their aids, or they programmed their aid incorrectly.

As a result of this field testing we will be revising the manual and incorporating the comments and suggestions provided by the participants. At that time, additional testing will be performed.

Another test was designed for the evaluation of the T-TAM functionality and reliability. A T-TAM, associated cables, and a manual was supplied to a disabled individual through a prospective manufacturer. A representative of the manufacturer with experience in devices of this type assisted the individual in the setup and programming of his electronic aid. Trace Center staff provided additional information when needed.

The T-TAM has been in place with this individual for more than three months with no reported breakdowns or major problems. While two specific requests were made for changes in the operation, they were minor in nature and not necessary to the effective operation of the T-TAM. Nevertheless, the feasibility of implementing these requests is currently under review. As a result of this initial success, additional field testing with other disabled individuals is being scheduled prior to commercial availability.

A third test was performed independently by a major computer manufacturer to determine compliance with FCC regulations on the emission of radio frequencies. This manufacturer provided suggestions on how to lower these emissions. These suggestions are being incorporated in a redesigned printed circuit board.

After incorporating recommendations and suggestions in the design of the T-TAM and the T-TAM manual, additional field testing will be performed under outside funding to determine the effectiveness of the T-TAM in meeting the needs identified earlier.

Refinement

Modifications to the T-TAM and manual have already been taking place based on information received from both internal and external evaluation.

The manual is being rewritten to include additional pictorial descriptions of how to connect the T-TAM to the particular computer. Sections are being added to show how to connect the T-TAM to several popular aids. Finally, specific step by step instructions on what the user has to program in their aid to type a specific key or to move the mouse are being expanded. It was felt that these modifications would result in a more effective manual.

A new printed circuit board is being designed to incorporate changes in the original circuit, to reduce electromagnetic emissions, and to allow for expanded programming needs. After this stage, the hardware design will be complete with no further modifications anticipated.

The software continues to be modified based on changes to the GIDEI standard and feedback from evaluators. The design of the T-TAM allows us to continue to modify the software, if necessary, throughout the life of the device.

Commercialization Efforts

The Trace Center has been in contact with two companies that have expressed interest in the manufacture, distribution and support of the T-TAM. The companies are The Prentke-Romich Company of Wooster, Ohio, and Words+, Inc. of Lancaster, California. These companies develop and market products for people with disabilities on a national level, and both have several years of experience in this field.

Prototypes of the T-TAM have been sent to these companies for evaluation and testing. We will be working with these companies to discuss any modifications as well as to provide technical assistance during the commercial transfer process.

At this time, manufacturers have estimated that the product will sell for under $500. We continue to work with them on lowering the cost of the T-TAM through lower parts cost and lower product support requirements.

Attachments

Grant Proposal: "Alternate Access Interface for Mouse and Touchscreen/tablet Based Computers and Software for Severely Physically Handicappeu Students"

Photographs of T-TAM Device

T-TAM Manual

T-TAM Schematics

T-TAM Software

Alternate Access Interface for Mouse and Touchscreen/tablet Based Computers and Software for Severely Physically Handicapped Students

I. Abstract

Access to standard computer systems and software by individuals with severe physical disabilities is important in order for them to compete in today's educational and employment settings. The advances in the development of computer systems have increased both their pow.r for processing information and the types of applications for which they can be used. Many advances designed to ease the use of the computer and to make them more "user friendly" (such as mice and touchscreens) have presented new barriers in the use of computer systems by people with severe physical disabilities.

In order for people with severe physical disabilities to use standard computer systems and software, they must be able to provide the same type of input as the new standard input devices. The concept of an Keyboard Emulator (a special interface module) which is able to simulate keyboard activity from input received from a special electronic aid used by a severely physically handicapped child was pioneered by the Trace Center over eight years ago and is now in common commercial availability from many manufacturers. However, these Keyboard Emulators do not work on some of the newer computer systems, and do not emulate the newer input devices such as the mouse, touchscreen, or touchpad.

The goal of this project is to create a new type of Emulator, the General Input Device Emulator (GID Emulator), which would work on the newer computers and provide a means to emulate all the standard input devices of these newer computers. The GID Emulator would allow a severely physically handicapped child who has an electronic aid, already designed to use a Keyboard Emulator, to use the new computers appearing in his educational environment. The use of the GID Emulator would allow this child to participate in all the computer based learning activities with all the other non-handicapped children.

Table of Contents

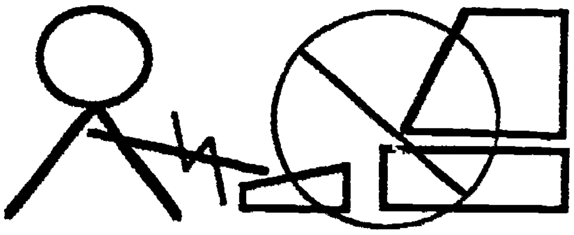
II. Introduction

A. Educational Problem to be Solved or Minimized

1. Original Problem

As the use of computers continues to become an integral part of the educational environment, the need for all children, who are physically and non-physically handicapped, to be able to use the school's computers is of great importance if those children are to have an equal opportunity for a quality education.

ORIGINAL
PROBLEM



Disabled Students were unable to use the keyboards on standard computers which are rapidly becoming in'egral components of standard and special classrooms

Figure 1

However, there are many physically handicapp :d children who are unable to use a computer because they cannot use the standard input devices (e.g. keyboard, mouse) as they come from the manufacturers.

2. Past Solution Strategies for the Moderately Handicapped Child

Solutions have been developed for moderately physically handicapped children to be able to use the keyboard. Keyguards (a plate mounted above the keys with holes over each key) have been made for many computer keyboards to assist children, who require a surface to steady their hand, to use a keyboard. Mechanical latches for the shift keys have been made to let single finger typists (or head stick and mouth stick typists) "hold" down the shift key while typing another key. Software modifications to operating systems have been developed to accomplish this same "Sticky Key" feature.

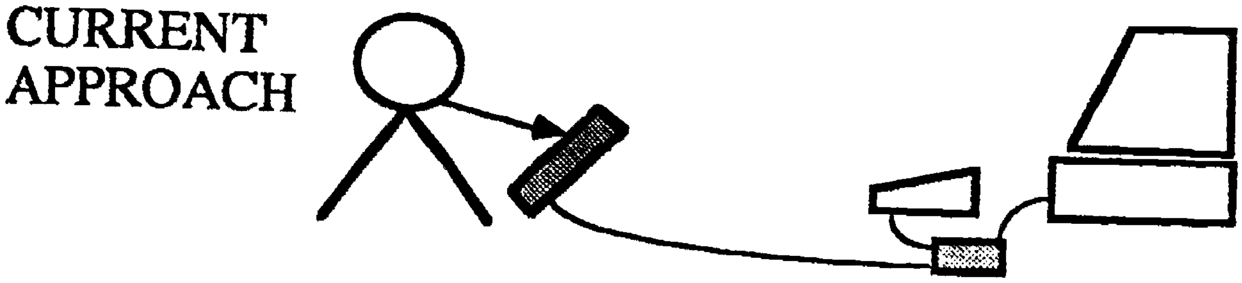
For the moderately physically handicapped child, who cannot use the mouse but can use the keyboard, a feature called Mouse Keys has been developed which allows the child to use the keyboard to simulate mouse motion and mouse button activ'ty.

However, there are still many severely physically handicapped children who are unable to use the standard keyboard at all. Some are too weak to press the keys. Some do not have the range of motion required to reach all the keys. Some do not have sufficient functional motor control of their hands, legs, or head to use the standard keyboard.

3. Past Solution Strategies for the Severely Handicapped Child

In addressing the communication needs of the severely physically handicapped child, there have been many different types of electronic devices developed that are specially interfaced to the abilities of the child. The aids may have smaller keyboards, or single switch input (e.g. eye blink switch, eyebrow switch, pedal switch), or two switch (e.g. sip-&-puff switch) Morse code input, or eyegaze input.

There are the SST Eyetyper and the Santec CEDRIC which allow the handicapped child to type by just looking at the "keys". The Words+ Equalizer allows input from the handicapped child using singe switch scanning or d ·l.switch Morse code. The Canon Communicator has a reduced sized keyboard fcr the handicapped child with limited reach. The ACS ScanPak allows the handicapped child to type using head pointing with single switch scanning or a headbeam lightpointer. The Autocom allows the handicapped child with severe cerebral palsy to type using a magnetic handpiece pointer and a large flat keyboard. The Prentke-Romich Express III allows input from scanning, headpointing, joystick, or slidepointing.



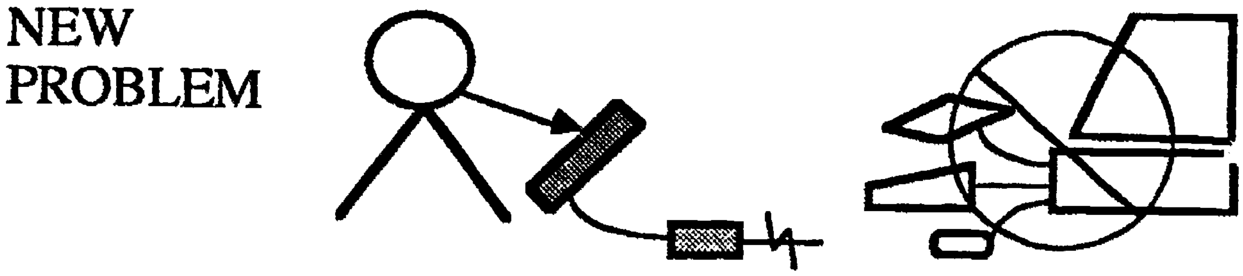
Special aids are connected to the computers through Keyboard Emulators which make input from the special aids look like it is coming from the computer's standard keyboard

Figure 2

Many of these electronic aids have been adapted to address the issue of computer access by using a device known as a Keyboard Emulator. A Keyboard Emulator is an electronic hardware device that allows the child's electronic aid to simulate (emulate) keyboard activity of the school's computer so that software programs running on the school's computer cannot tell the difference between the keystrokes from the real keyboard, and those from the child's "keyboard". The computer's real keyboard and the child's electronic aid are connected to the Keyboard Emulator so that both the real keyboard and the child's "keyboard" can be used at the same time. With the Keyboard Emulator, no modification to the software program running on the computer is required.

4. Problems with the Current State of Keyboard Emulators

At this time, these Keyboard Emulators are available commercially from a wide number of vendors (Prentke-Romich, TASH, ACS, Zygo, Words+, etc.) but for only a limited number of computers (IBM PC, Apple II+, Apple IIe). One of the reasons for this limited number is that, in the past, each computer model needed a different Keyboard Emulator because of electronic differences in the way the keyboard works. For example, a different Keyboard Emulator is required for the Apple II+, the IIe, and the IIc. Similarly, different Emulators are needed for the IBM PC, AT, new AT, and PS/2 series keyboards as well. Another reason is the amount of research and development time required to create a Keyboard Emulator. New models of computers have been coming out too fast for the rehabilitation engineers to keep up.



1. Current keyboard emulators do not work with the newer generation of computers.
2. Current keyboard emulators only address keyboard input leaving the disabled child with no access to mouse or touchpad input required by the new computers and software.

Figure 3

In order to alleviate the need for Keyboard Emulators to be implemented in hardware, work is being done to provide Keyboard Emulators as software modifications to the operating system (or as programs running concurrently in a multitasking environment). These software Keyboard Emulators would be able to take input from the electronic aids and "inject" keystrokes into the operating system. This makes it possible to produce Keyboard

Emulators more quickly, since electronic differences in the keyboards require little if any change in the software patch, and less expensively since no hardware is involved.

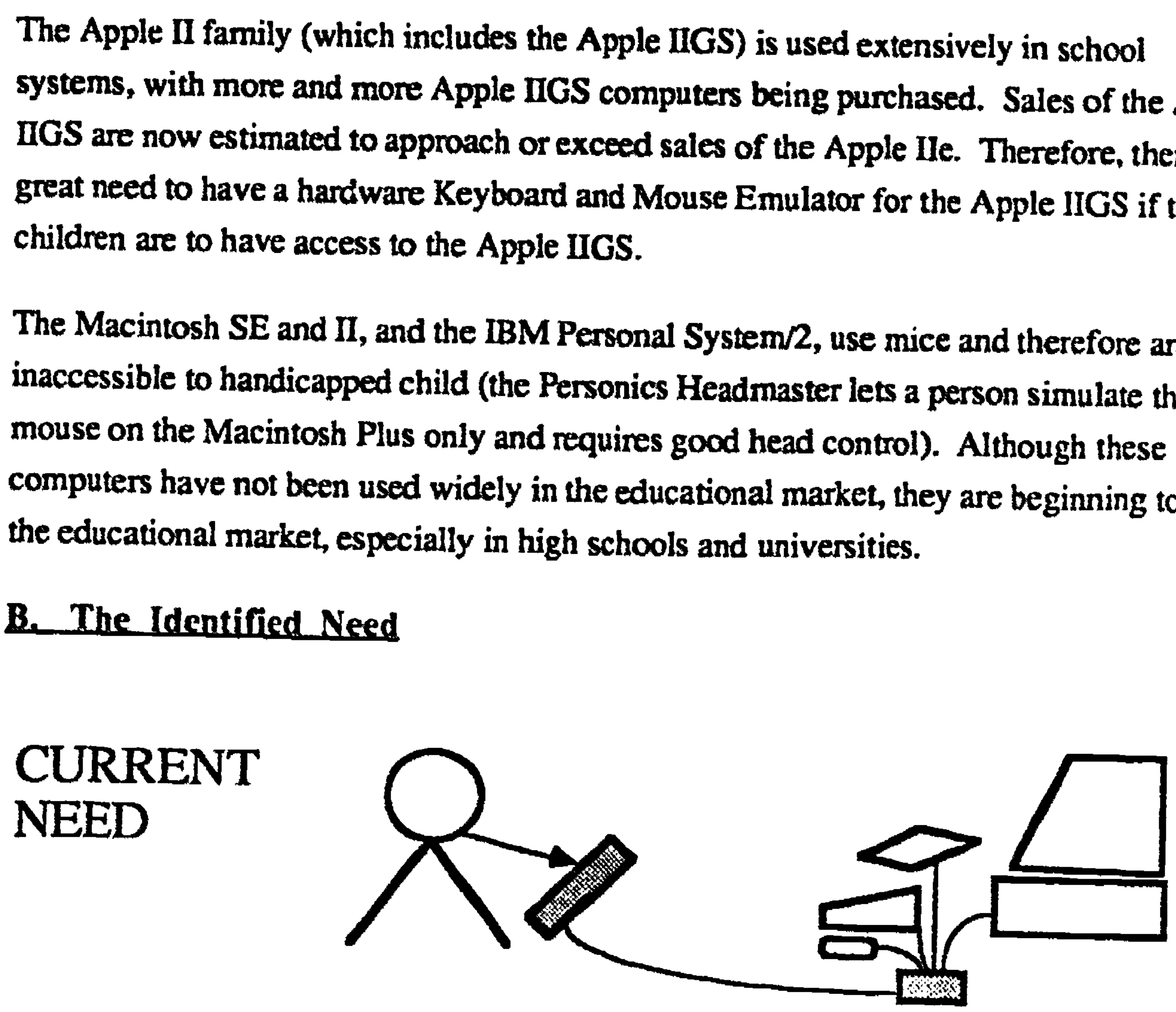
However, there are still circumstances where it is not possible to create software Keyboard Emulators because of the design of the operating system. One such computer is the Apple II family, including the IIGS. The Apple IIGS must have a hardware Keyboard Emulator when it runs Apple DOS (there are many programs that are used in the educational curriculum which run under Apple Dos).

There is a way to use the Keyboard Emulators designed for the Apple IIe with the Apple IIGS but the Keyboard Emulator cannot simulate mouse activity, and many of the children who must use an electronic aid also cannot operate the mouse.

The Apple II family (which includes the Apple IIGS) is used extensively in school systems, with more and more Apple IIGS computers being purchased. Sales of the Apple IIGS are now estimated to approach or exceed sales of the Apple IIe. Therefore, there is a great need to have a hardware Keyboard and Mouse Emulator for the Apple IIGS if these children are to have access to the Apple IIGS.

The Macintosh SE and II, and the IBM Personal System/2, use mice and therefore are also inaccessible to handicapped child (the Personics Headmaster lets a person simulate the mouse on the Macintosh Plus only and requires good head control). Although these computers have not been used widely in the educational market, they are beginning to enter the educational market, especially in high schools and universities.

B. The Identified Need



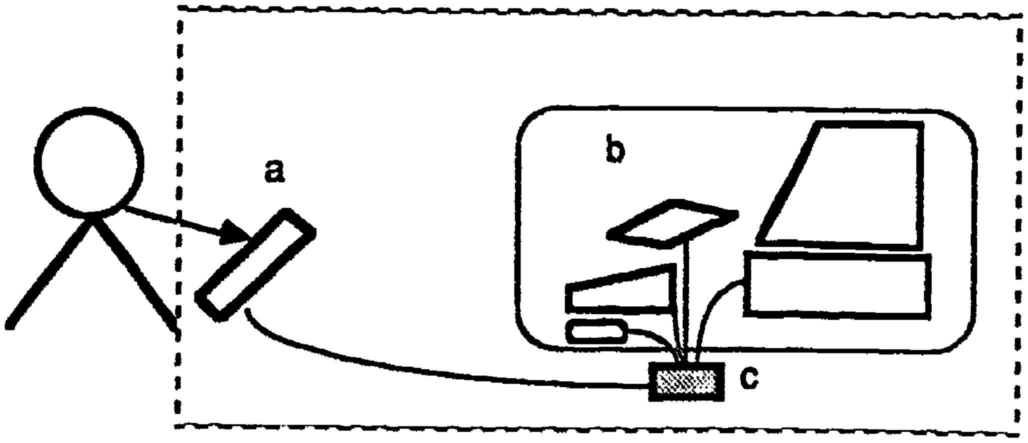
A new type of adapter module is required, one which can emulate the input of the newer computer's standard mice and touchpads as well as the keyboard. Such an adaptor would be a General Input Device Emulator.

Figure 4

The identified need is two-fold. First, Emulators must be provided for the Apple IIGS and other new computers for which no Emulator currently exists. Second, these new Emulators must be able to handle more than just keyboard input. They must also simulate the new input devices such as mice, and touchpads or touchscreens.

C. The Proposed Solution

The proposed solution is to create an accessible computer system by developing the missing link between the electronic aids and the Apple IIGS computer. This missing link is a General Input Device Emulator (GID Emulator), a device that connects between the aid and the computer which would allow the aid to send commands to the GID Emulator in a standard protocol which would mimic the input from the computer's standard input devices (e.g. keyboard, mouse, touchpad, touchscreen).



The proposed system is an accessible computer system which is composed of:

(a) commercially available special input devices,
(b) standard school computers with their input devices,
(c) and a special General Input Device Emulating interface module.

Figure 3

Commercially available electronic aids will be purchased which already support the Keyboard Emulating Interface (KEI) Standard. The aids will then be enhanced to add the new features described below in "Additional Protocols for Mouse and Touchpad Input" by enhancing the KEI software routines. This will be done in cooperation with, and assistance from, the manufacturers of the aids. The GID Emulator will then be constructed and tested using the enhanced aids.

The GID Emulator will have the following features:

1) Compatibility with the Keyboard Emulating Interface Standard

The GID Emulator will be a superset of the current Keyboard Emulating Interface (KEI) Standard. This will ensure that many aids will be able to use the GID Emulator right away. After time, with upgrades of aids, the aids will be able to take advantage of the new features described below as well.

2) Additional Protocols for Mouse and Touchpad/Touchscreen Input

In order for this alternate access strategy to work with the new computers, additional protocols for relative motion input devices (e.g. mouse) and absolute position input devices (e.g. touchpad, touchscreen) will be part of the GID Emulator Interface Standard. This will allow physically handicapped children who are unable to manipulate a mouse, use a touchpad, or reach the touchscreen to be able use programs that require input from these devices.

3) Low Cost of Manufacture

The design of the GID Emulator will be done to minimize the cost to manufacture the device. Parts used in the design will also be chosen so that they are readily available from several sources.

4) Parallel Operation with Real Input Devices

Use of the GID Emulator must not disable the use of the normal input devices (e.g. standard keyboard, mouse, etc.). This is necessary so that interference with the "normal" use of the computer is avoided. This feature also facilitates instruction and demonstration of the software programs by the teacher to the child where the teacher and child must take turns using the input devices.

5) Infrared Link (optional)

This feature would eliminate the need for the teacher or an assistant to connect the child's electronic aid to the GID Emulator each time the child wanted to use the computer. Instead the communication between the aid and the GID Emulator would take place through the infrared light link. This link might be built in to the GID Emulator or might be an accessory that is attached to the electronic aid and the GID Emulator.

The proposed system consists of a commercially available electronic aid, a standard school computer with their input devices, and the GID Emulator. The eventual cost of the custom components of the system would be targeted to be under 10% of the cost of the aid selected and the standard school computer. The enhancement to the aid to support the GID Emulator Interface Standard by the manufacturers could bring the cost down to 1-3% of the system.

D. Complexities Involved in the Proposed Solution

Although the objective of this project is to create a simple solution to the problem, the issues involved are somewhat complex. A simple solution, in this case, would be the best solution primarily because it is the best way to contain the cost. A simple solution is also generally more flexible and more straightforward, requiring less training time.

The issues involved in this simple (minimal) solution, however, are not straightforward. New input systems are requiring finer control on the part of the user. This finer control is the opposite of the type of control that severely physically handicapped children have. Thus, this project must find a simple way to allow individuals with severe motor control problems to use the computers whose newer input systems require fine motor control.

In addition, this must be done in a way that is technically compatible not only with the new microcomputers and their operating systems, but also with the existing special interface aids (Prentke-Romich Express III, Words+ Equalizer, Zygo Tetrascan, ACS Speechpak, etc.).

Finally, if the individual is to have access to more than one single make and model computer, then some type of standard communication protocol needs to be developed which will work across different makes and models of computers as well as operating systems.

E. Need for Assistance

Assistance is needed in developing this crucial link because of the high cost associated with developing a hardware GID Emulator. Research and development costs, if borne by the manufacturer, are inevitably passed on to the consumer in the form of a high cost for the adapter. One might argue that the cost of the GID Emulator, although possibly high, is only a small fraction of the cost of the expensive electronic aids purchased by the child. However, the GID Emulator will not necessarily be purchased by the child (or the child's funding source), but most likely by the school in order to increase the accessibility of their computers. The cost of the GID Emulator must be low in order for schools to be able to purchase them.

III. Objectives

A. Principal Objective

The principal objective of this proposal is to provide an accessible Apple II family of computers through the development of a General Input Device Emulator for the Apple IIGS. The Apple IIGS is selected since 1) the Apple II is the most widely used family of computers in the elementary and high school settings, and 2) the Apple IIGS is the model being used to develop the next generation of special and regular education software.

B. Subordinate Objectives

A subordinate objective would be to create the General Input Device Emulator Interface (GID Emulator Interface) Standard. This standard would assist those involved in creating software Keyboard Emulating Interfaces to create software GID Emulators, and to assure that they would be compatible with each other and with our hardware GID Emulators. A standard is crucial since electronic aid manufacturers need to be assured that if they design their aid to support the GID Emulator, then :hey will work with GID Emulators for different computers.

C. Secondary Objectives

A secondary objective would be to provide General Input Device Emulators (either in hardware or software) for the Macintosh SE and II, and the IBM PS/2 line of computers, if time allows.

Although these computers are not readily used in the elementary and middle school at this point, there are two reasons for creating GID Emulators for these computers. First, we do not want to deter schools from buying different types of computers just because they are inaccessible to the physically disabled child.

Second, there are many other individuals who need to have access to computers who are in an educational setting which use these computers (e.g. universities, colleges, technical schools, high schools). Those who are physically disabled need to be able to continue their education and use all of the computers that are encountered, even past elementary school.

Creating a GID Emulator for the Macintosh SE and II would be relatively easy since they use the Apple Desktop Bus (which is also used on the Apple IIGS) for keyboard and mouse input. Apple Computer is standardizing their input devices on the Apple Desktop

Bus which means that the GID Emulators created for the Apple IIGS are likely to work for the Macintosh SE and II, and future Apple computers.

IBM is also making an effort to standardize their keyboard and mouse input devices. The IBM PS/2 line of computers all have the same keyboard and mouse input port so creating an additional GID Emulator for the IBM PS/2 line would greatly increase the number of computers for which alternate access is available.

IV. Results or Benefits Expected

The benefit of having GID Emulators available for the Apple IIGS (as well as the Macintosh SE and II, and the IBM PS/2 line of computers) is that the school districts will not be deterred from buying new and more powerful computers because of the fear of leaving the physically disabled children behind since they will not be able to use these new computers. Allowing the schools to freely upgrade to new computers and new software will enhance the ability of the educational program to make use of improving technology.

The availability of the GID Emulator for these computers would encourage more development of electronic aids to utilize the GID Emulator. Many manufacturers and researchers look to the Trace Center to provide the emulators since we have the technical expertise and are the authors of the Keyboard Emulator Interface Standard. To create new emulators is often beyond the capabilities and resources of aid manufacturers and researchers.

A side benefit would be to allow the physically disabled individual to be more competitive in employment since they would be able to access an increased number of computers found in the work environment with the same electronic aid which they used in school.

V. Approach

A. Outline of Plan of Action and Accomplishments to be Achieved

The plan of action will be split into five tasks listed below:

Task 1: Substantiate the Usability of the GID Emulator's Design
Task 2: Create the GID Emulator
Task 3: Field Test the Aid and Report on Field Test Results
Task 4: Prepare Marketing Recommendations
Task 5: Evaluation of the Product/Project

For each task, quantitative projections of the accomplishments to be achieved will be stated. The data to be collected, maintained, and evaluated, and the criteria for evaluating the results and successes of the project will be covered in Task 5.

Task 1: Substantiate the Usability of the GID Emulator's Design

The purpose of this task is to verify the assumptions underlying and characterizing the proposed GID Emulator. This will involve manufacturers and school systems. Discussions with them will be held in order to verify the exact nature of the need and the constraints involved.

Discussions with prospective manufacturers of the GID Emulator will be held to determine the marketing and manufacturing methods, the time to market for new products, and other limitations that might exists. Reaction will be solicited concerning the proposed solution.

Preliminary discussion with some prospective manufacturers indicate that:

1) they are very interested in a GID Emulator,
2) they are able to produce complex electronic devices,
3) they assemble products upon order,
4) they keep less than three months sales in stock,
5) common parts reduce the cost of the inventory,
6) and it takes 2-3 months to take a finished product and bring it to the market.

Discussion with teachers and administrators at local schools (and other school districts) will be held to identify the extent of current problem areas, the plans for computer usage in the education program, and the number and types of computers that are projected to be bought for the next five years. Reaction will be solicited concerning the proposed solution.

Preliminary discussion with local school districts indicate that:

1) the elementary and middle schools are and will be buying predominantly the Apple IIGS,
2) the high school system will be changing over from using the Apple IIe to the Macintosh SE,
3) the school system will be more than doubling the number of computers used in school curriculum in the next five years,

4) and adaptive equipment to make computers accessible by handicapped children must be less than $500 to be considered for purchase by the school system.

Task 1 will be completed by the first month.

Task 2: Create the GID Emulator

The first step will be to become very familiar with the electronics, protocol, and software involved in the Apple Desktop Bus (ADB). Features of the GID Emulator will be specified based on the results of Task 1.

Second, a study of the new input systems (e.g. mice, touchpad) will be performed to determine their characteristics, both within computer systems and between different computer systems. The operating systems will be examined to determine how each of the different operating systems treat these types of inputs. Application programs will be examined to determine what type of action are expected to be performed by the user with these new input systems.

Third, a preliminary conceptual design will be made. The conceptual design will allow for meeting the design goals set forth in Task 1. The design will also allow sufficient modularity to make designing GID Emulators for other computers (Macintosh SE and II, and the IBM PS/2 line) require minimal new or redesign. Modularity can also reduce the cost of the inventory that a manufacturer must maintain in order to meet the demand. In our preliminary discussion with manufacturers, they indicated that a small inventory is crucial. A programmable single chip processor will most likely be the center of this design.

The fourth step is selection of the hardware components and making a preliminary design. Selection will be based on:

1) keeping them as simple as possible, yet fulfilling the needs of the design,
2) using off-the-shelf components that have future widespread availability,
3) selecting parts that have more than one supplier,
4) and cost.

This preliminary design will be sent out for review by potential manufacturers of the GID Emulators for their input on the cost of producing such a device based upon their current capabilities. We will also have made arrangements for Apple and IBM to review the preliminary design for compatibility with their products.

After receiving the feedback from potential manufacturers, a modified working design will be drawn up and a prototype made and alpha tested in the lab. After successful alpha testing, a duplicate unit will be produced to be given to the funding agency. Three to four more units will be produced to be used in the field testing of the device, and will later be given for evaluation to prospective manufacturers.

Concurrent with the above activities, a document will be prepared that describes the GID Emulator with regard to the components used, the component costs, the supply sources, any proprietary issues concerning each component used, and enough technical information to allow reproduction of the GID Emulator. A user's guide will also be prepared that describes how to use the GID Emulator. In order to facilitate its use by persons with physical disabilities, the manual will be also provided in an electronic form as a text file on a disk.

This task will be completed by the 12th month in order to allow sufficient time for task 3.

Task 3: Field Test the Aid and Report on Field Test Results

Five to ten field test individuals will be identified in Madison and the surrounding areas who have aids which are similar or identical to the aids used (or could be easily modified to work with the GID Emulator). These individuals will be selected during the first six months of the project.

The field test will consist of placing the GID Emulator with the child for a test period of a week. On the first day, they will be brought to the Trace Center and shown how to use an Apple IIGS (or Macintosh SE and II, and IBM PS/2 family) and show how to accomplish the 5 tests that will be asked of them after the end of the week period. If they do not have access to an Apple IIGS, a computer will be lent to them for the week period. They will have the week to learn how to use the GID Emulator to accomplish the 5 tasks, assisted only by the users guide.

The 5 tests will be designed so that a combination of keyboard, and mouse, or touchpad input will be required. They will be selected to represent a cross section of the kind of activities that might occur in special or regular educational software. They will also involve many activities that are common when using the computer. A list of possible tests are listed below:

Test 1: Open a folder, and run a program.

Test 2: Move objects around on the screen to assemble the picture of a car.

Test 3: Fill in blanks and correct errors in sentences presented on the screen

Test 4: Select items presented on the screen by a touchscreen program..

Test 5: Write or trace their name in cursive writing using a draw or paint program.

This preliminary version of the field test plan will be drawn up and submitted for review by the clinicians, teachers, and children involved. After review, the final field test plan will be modified as required. The field tests will begin as soon as the beta test units and user guide are completed from task 3.

Task 4: Prepare Marketing Recommendations

From work done in Task 1 and Task 2, a marketing recommendation will be drawn up.

Prospective manufacturers identified in Task 1 will each be given a GID Emulator and a GID Emulator Interface Standard document after alpha testing is complete. At the end of the beta test period, those manufacturers that have indicated a commitment to marking the GID Emulator will be given full documentation and a final post-beta test GID Emulator. We will assist each manufacturer as needed. Based upon past work with manufacturers and preliminary discussion on this project, it appears as if at least one and probably multiple manufacturers would pick up this Emulator if it were developed. In keeping with past practices, we prefer the non-exclusive release of products to manufacturers to maintain competition and spur further product development. Our record to date in this area is very good with approximately 95% of all products developed by the Trace Center becoming a commercially product.

We will also encourage electronic aid manufacturers to include the GID Emulator with their aid. Electronic aid manufacturers will be given a copy of the GID Emulator Interface Standard document so that they can begin to design their electronic aids to take advantage of the full capabilities of the GID Emulators.

Our marketing plans address patent, copyright, and other regulatory or proprietary issues which affect the marketing of the aid. Our preference is non-exclusive license and zero-royalty to keep costs to consumers and schools at a minimum.

We hope to have a commitment from from a manufacturer to manufacture the GID Emulator by the end of the beta testing (18th month) and to have the GID Emulators commercially available by the end of the 24th month.

A paper on the device will be written up for, and submitted to, the Rehabilitation Engineering Society's (RESNA) annual conference proceedings and the Closing the Gap conference on Computers and the Disabled in Minnesota.

Task 5: Evaluation of the Product/Project

The purpose of this task will be to provide a concrete means to evaluate the success of this project based upon the goals set forth in Task 1.

The evaluation of success will be done in three areas: technical, student/teacher, and manufacturer/marketing success:

1) Success from the technical standpoint is if the GID Emulator was able to completely emulate the input from the standard input devices. This will be tested by demonstrating the ability to operate all functions of a cross section of software written to use the standard keyboard, mice, and touchpad/touchscreen for the target computer(s).

2) Success from the student and teacher's standpoint is if the GID Emulator actually solves their access problem without introducing new problems. This will be tested by the completion of the five tests on the computer which require use of the keyboard, mouse, and touchpad, which represents the spectrum of required inputs that are found in a classroom. An average completion rate of 4.8 out of 5 or better would be considered successful.

3) Success from the manufacturer's standpoint is if they indicate satisfaction with the marketability of the GID Emulator. Acutal commercial production of the device is the measurable criterion in this area.

Overall project success is a function of all three areas. The project would not be deemed successful unlesss all three criterion were met. This 1) provides full access to standard mouse, and touchpad operated software. 2) can be operated successfully by disabled children using special interface aids, and 3) it is commercially picked up and produced.

B. Personnel, Facilities, and Consultants

The Trace Research and Development Center is internationally known for its leadership in the area of computer applications for the disabled individual. The center maintains a balance of research, d^evelopment, information dissemination and clinical services. The interdisciplinary staff of the Trace Center provides a solid background in both the needs of the disabled individual and the capabilities of technology. The Trace Center includes both engineering and experimental laboratory facilities for support of this project. The engineering facilities have a proven capability to design and fabricate electronic equipment for both research and commercial dissemination. The extensive experience of the Trace Center staff offers an in-house pool of specialist for consultation on any equipment or experimental design problem.

In addition, the Trace Center has extensive experience in the area covered by this proposal. Trace pioneered the original concept of Keyboard Emulating Interfaces, developed the Keyboard Emulating Interface Standard, and designed KEI's for the Apple IIe, IBM-PC, and DEC VT100 terminals. The Trace Center is currently serving as headquarters for the Industry/Government Initiative's Task Force on Computer Accessibility which is studying the problem of mouse and touchscreen input.

Trace has also had experience in the Technology Compensatory Activities Program. Trace was a participant in the original program under which we developed the Trine Communication/Writing/Computer Access aid. This aid was subsequently picked up and commercially distributed by two manufacturers.

The Trace Center Staff who will be involved in the project are:

Project Staff

Name	Position	Year 1 %FTE	Year 2 %FTE
Gregg Vanderheiden	Center Director, Principal Investigator	10.0	5.0
Charles Lee	Project Manager, Lead Investigator	30.0	15.0
Joe Schauer	Engineer	30.0	7.5
Julie Gamradt	Clinician	10.0	5.0
Peter Borden	Writer	10.0	10.0
Staff	Secretary	25.0	17.5
Staff	Lab Technician	20.0	5.0
Staff	Project Assistant	20.0	25.0

Person Loading Chart

Activity	GV	CL	JS	JG	PB	LT	PA
Verify GID Emulator's Usability	3	5	1				
Investigate electronics of ADB		5	4				
Investigate protocol of ADB		2	4				
Make a preliminary design	2	6	7				
Review preliminary design	2	2	1				
Make final design		2	5				
Construct GID Emulator			3			15	
Alpha test the GID Emulator	1	1	2	2		3	
Make design revisions		2	5			2	
Construct revised GID Emulator			2			5	
Prepare product documentation	1	2	1		5		
Prepare user's guide	1	2			10		
Select subjects for field test	1	2		3			10
Prepare field tests	1	2		2			5
Perform field tests		1		6			30
Wnte up results of field tests		1		2	5		10
Co ısult with manufacturers on marketability	1	5	1				
Assist manufacturers on commercialization	2	5	1				
Total	15	45	37	15	20	25	55

Consultants

Prentke-Romich is currently manufacturing and marketing the Key board Emulators for the Apple II+, IIe and the IBM PC. They are also a nationally and internationally known electronic aid manufacturer.

Words+, Inc. is currently marketing the Trine System, an electronic aid, which was developed by the Trace Center. Words+, Inc is an internationally known electronic aid manufacturer.

Madison School District has an exemplary program in mainstreaming of disabled children.

C. Biographical Sketch of Key Personnel

Gregg C. Vanderheiden, Ph.D.

Dr. Vanderheiden is the director of the Trace Research and Development Center, and a professor in the Department of Industrial Engineering at the University of Wisconsin - Madison. He is an internationally known expert on the application of computer systems for the disabled. As the founder of the Trace Center, he has been a leader in the research and development of both specialized computer systems and access to standard computer systems for the disabled individual. He is currently serving as co-chairman of the Design Considerations Task Force of the Industry/Government Cooperative Initiative On Computer Accessibility. He also consults with several of the largest computer manufactures on how to develop and modify computer systems to be accessible by the disabled individual. Dr. Vanderheiden is the Co-Principal Investigator on the project,. He will focus his time on the project specifications and testing portions of the project as well as general project monitoring.

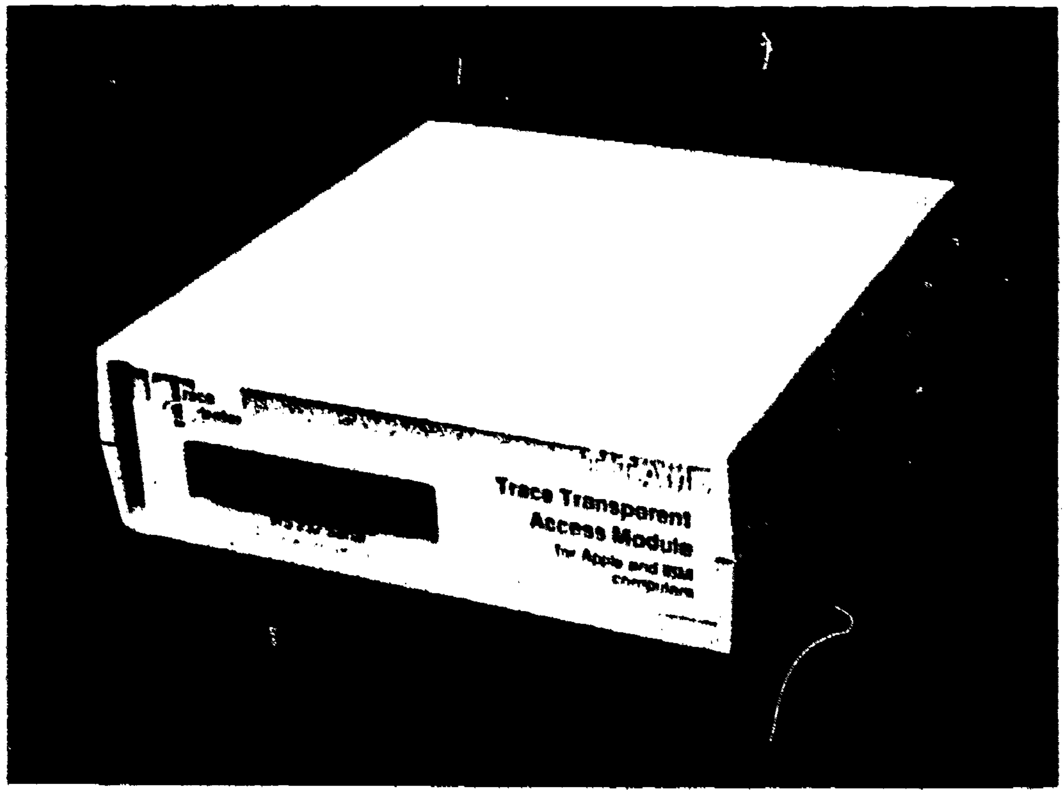
Charles C. Lee, MS ME

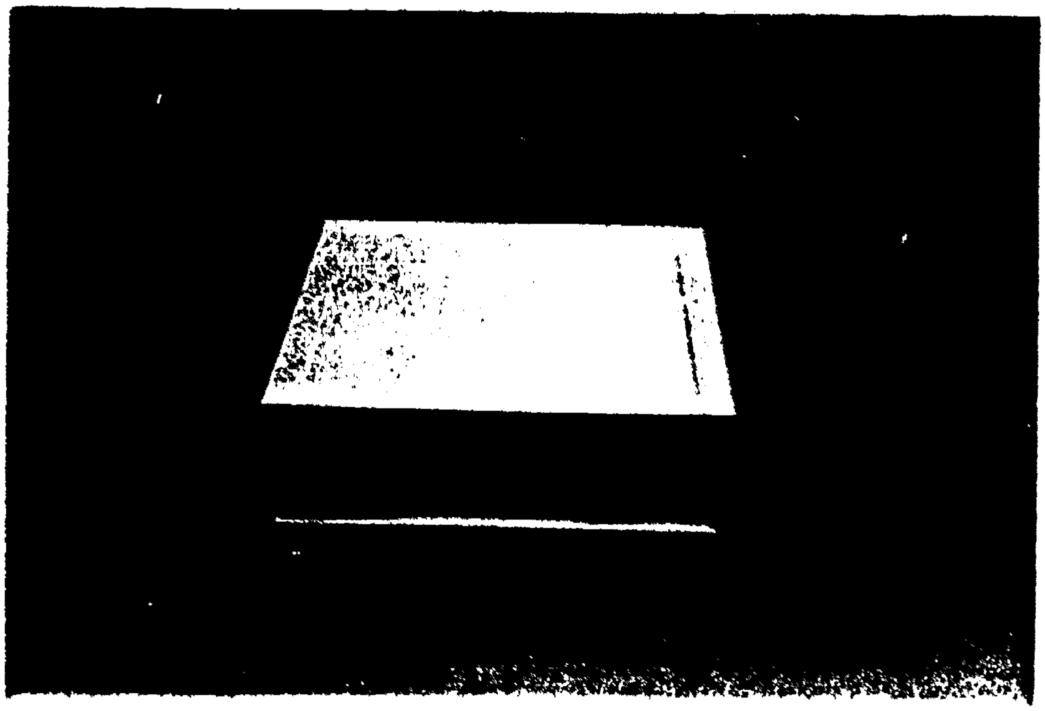
Mr. Lee is a full-time staff member of the Trace Research and Development Center. He graduated from Stanford University with a Masters from the Mechanical Engineering Department in Engineering Design, with an emphasis on Smart Product Design. He has been with the Trace Center for three years, during which time he has developed several products to increase the accessibility of computers for the disabled individual. He has written a program that allows one-finger operation of the keyboard for the IBM PC, IBM XT, IBM PCjr, IBM Convertible, and IBM AT and true compatibles. This program is commercially available through the Trace Reprint Service. He has also written a program that provides abbreviation/expansion input acceleration for the IBM family of computers. This program is currently in the process of commercialization. He is the Coordinator for the "Consideration in the Design of Computers and Operating Systems to Increase their Accessibility to Persons with Disabilities" document which is a product of the Design Consideration Task Force of the Industry/Government Cooperative Initiative On Computer

Accessibility. Mr. Lee is the lead investigator for this project. As such, he will oversee the daily operation of the project and maintain the management plan for the project. He will also be extensively involved in the technical design of the GID Emulator.

Joseph Schauer, BS EE

Mr. Schauer is a full-time staff member of the Trace Center. He graduated from the University of Wisconsin - Madison. Mr. Schauer has been with the Trace Center for two years and is currently the project manager for the Keyboard Emulating Interface Standard and for the software Keyboard Emulators that are being developed here at Trace. He has written a software Keyboard Emulator for the IBM PC family of computers. He was also involved in the creation and debugging of the micro-processor based Long Range Optical Pointer peripheral card for the IBM PC, the dual keyboard input for the IBM PC, and the multi-input box for the IBM PC. Mr. Schauer will be the primary person in the actual production, design, and fabrication of the GID Emulator.


Trace Transparent
Access Module
for Apple and IBM
computers



Trace Transparent Access Module (T-TAM)

for

Apple and IBM computers

Developed by
The Trace Research and Development Center
Madison, Wisconsin

Beta Version 1.0
May 1989

INTRODUCTION

What does it do?

The Trace Transparent Access Module (T-TAM) allows an individual who cannot use the mouse or keyboard to use their communication aid (or a second computer) to operate the computer's standard keyboard or mouse. Because the T-TAM is fully transparent, the computer cannot tell that the keystrokes and mouse movements coming from the person's communication aid are not coming from its standard keyboard.

Which computers does it work with?

The T-TAM works with Apple computers which use the Apple Desktop Bus (ADB) for their input devices. These include the Apple Macintosh II, IIx, IIcx, SE, SE/30 and Apple IIGS computers.

The T-TAM also works on IBM PS/2 Computers or compatibles. The T-TAM can be made to also work with IBM AT's and compatibles using two adaptor cables (and a PS/2 compatible mouse card, if mouse use is desired).

What software does it work with?

Because the T-TAM does its emulation of the keyboard and mouse in hardware, it is fully transparent. This means that it will work with *all* software and operating systems for the computers listed above. What software does it work with?

How do I learn how to use it?

This manual provides instructions on how to set up the T-TAM. It also explains what data needs to be sent from your communication aid to the T-TAM in order to successfully perform the keyboard and mouse functions on the computer. The manual is in five parts:

1) Getting Started: Covers how to connect the T-TAM.

2) Beginner Level: Covers the basic typing and mouse movement functions. All users should read and understand this section before proceeding to Intermediate and Advanced Levels.

3) Intermediate Level: Covers more sophisticated (but commonly necessary) typing and mouse functions.

4) Advanced Level: Covers functions not all users will need to know how to use.

5) General Reference: Lists all the commands used in operating the T-TAM. This reference will be valuable to *all* users of the T-TAM, once they have read sections 2 and 3 and are working on programming their communication aid to work with the T-TAM.

6) Technical Reference: Describes advanced programming techniques, and gives detailed technical information on how the T-TAM works. This section won't be necessary to most users.

GETTING STARTED

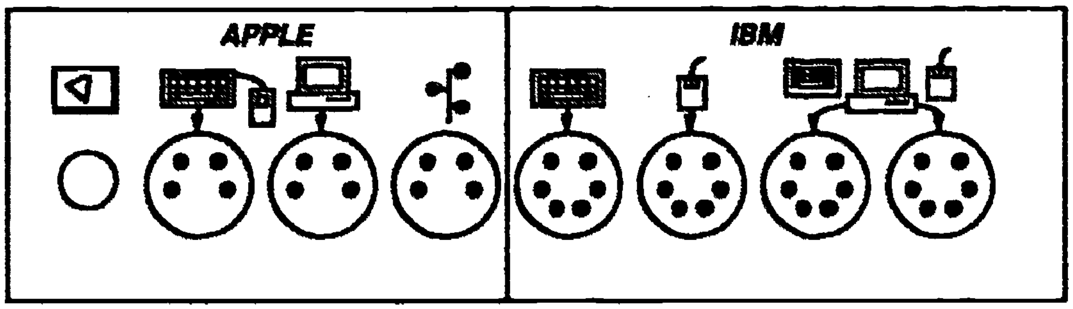
The Trace Transparent Access Module (T-TAM) can only work with one computer at a time. It has been designed to allow you to have two computers hooked up to it at a time. However, it will only work with the computer which is turned on, and only one computer should only be turned on at a time.

Follow the instructions below for the computer that you want to use.

Setting Up the T-TAM with Apple Computers

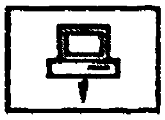
The T-TAM works with Apple computers which use Apple Desktop Bus (ADB) keyboards and mice (any Apple computer where the mouse can plug into the keyboard). Follow the steps below to properly connect up the Module. The instructions below deal with the connectors on the Apple portion (left side) of the back panel of the Module.

1. Turn off the computer.
2. Disconnect the keyboard and mouse from the computer.



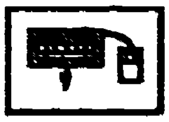
Back Panel

3. Disconnect your keyboard cable from your keyboard, and connect it to the T-TAM at the connector labelled with the symbol shown here.

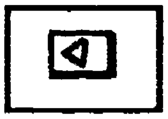


Reconnect the other end of the cable to the ADB connector on the computer.

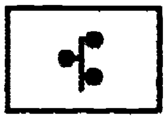
5. Take the short ADB cable that is provided with the T-TAM, and connect it to the T-TAM at the connector labeled with the symbol shown here.



6. Take the other end of the short cable and connect it to the keyboard.
7. Connect the mouse to the keyboard.
8. Some Macintosh computers (the Mac II family) allow you to turn on the computer using a key on the keyboard. You can perform a similar function with the T-TAM, by connecting a single switch to the connector labeled with this symbol.



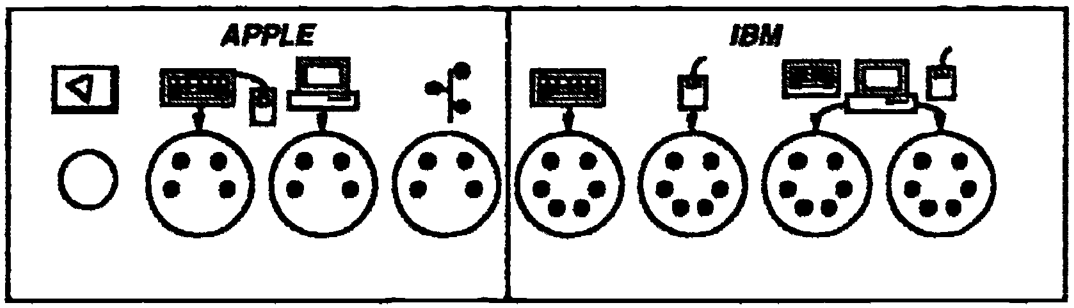
9. Any other ADB device can be connected to the connector labelled with the symbol shown here.



Setting Up the T-TAM with IBM computers

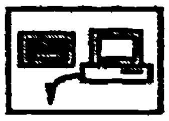
The T-TAM works with IBM PS/2 and AT computers and compatibles. Follow the instructions below to properly connect the T-TAM to the computer. All the instructions below deal with the connectors in the IBM portion (right side) of the back panel.

1. Turn off the computer.
2. Disconnect the keyboard and mouse from the computer.

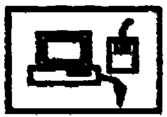


Back Panel

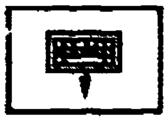
3. Take one of the long cables provided and connect it to the T-TAM at the connector labelled with the symbol shown here.



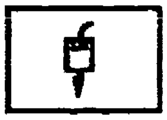
4. Connect the other end of the cable to the computer's keyboard connector (where the keyboard was connected).
5. Take the other the long cable provided and connect it to the T-TAM at the connector labelled with the symbol shown here.



6. Connect the other end of the cable to the computer's auxiliary connector (where the mouse was connected).
7. Connect the keyboard to the connector labelled just with the symbol shown here.



8. Connect the mouse to the connector labelled just with the symbol shown here.



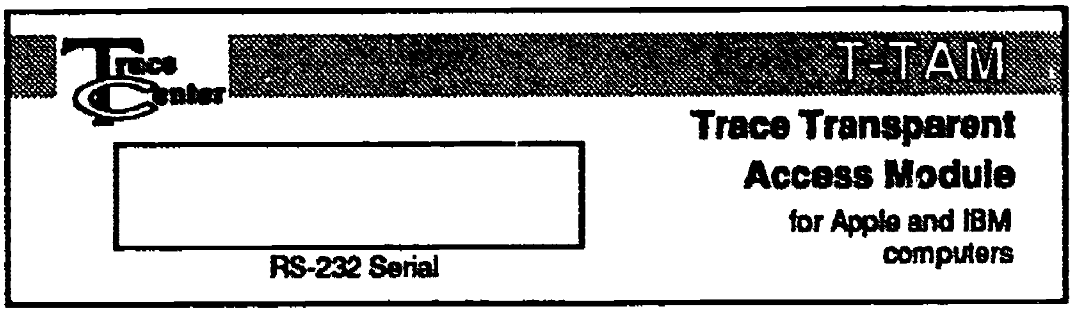
Setting up the Alternate Access Aid

The T-TAM works with an Alternate Access Aid or Communication Aid. From now on, we will refer to both types of aids as simply the Aid.

> NOTE: The technical details that follow are important for correct operation of the T-TAM. If you don't understand them, contact someone who can help you. It may save you trouble later.

In order to communicate with the T-TAM over the serial port, the Aid must be set up to communicate over the serial port. The settings should be as follows: 300 baud, 8 data bits, 1 stop bit, and no parity. The T-TAM is a Data Communication Equipment (DCE) device; therefore, the Aid must be a Data Terminal Equipment (DTE) devices. This means that you *may* need a null modem in order to hook up your Aid.

The T-TAM uses hardware handshaking in order to regulate the flow of data back and forth. Therefore, your Aid must support hardware handshaking or you may lose some data during transmission of large amounts of data at a high rate. The T-TAM is able to buffer up to 255 characters at a time.



Front Panel

Connect the Aid to the connector on the front panel panel labelled "RS-232" with a serial cable (you must provide this cable). If your Aid supports hardware handshaking, make sure pins 4 and 5 on the cable are also connected.

BEGINNER LEVEL

How the T-TAM Operates

The T-TAM accepts certain data from your Aid, converts it into data that the computer can "understand," and passes that data along to the computer, allowing you to control the keyboard and mouse functions of the computer.

The data sent from your Aid to the T-TAM is in a form called "serial ASCII." "Serial" simply means that the data is sent in a stream, one item at a time. "ASCII" stands for American Standard Code for Information Interchange. It simply refers to a standard way of representing letters as digital codes. The instructions below will refer to "ASCII characters." An ASCII character is a single letter, number, or symbol represented by a digital code. The letter "A," for instance, is represented by a particular number.

This information is just provided as background. You don't have to know what numbers represent what letters in ASCII. You just have to know what ASCII characters (A, B, C, 1, 2, 3, etc.) need to be sent to the T-TAM from your Aid in order to perform the keyboard and mouse functions you want to perform.

Notation in These Instructions

For the descriptions of how to use the T-TAM, the following notational convention is used:

`<esc>`	The escape character
`<null>`	The null character
`<###>`	character number ###
`<`*text*`>`	character called *text*
`A,B,C,1,2,3,%,$, etc.`	Printable ASCII characters

If you see a series of commands separated by commas and ending with a period, remember that the commas and periods are part of the command, and need to be sent from your Aid to the T-TAM as part of the whole command.

Simple Typing

Simply typing text using the T-TAM is fairly easy. All you need to do is to send ASCII characters to the T-TAM. Most communication aids already do this when you send a message that you have constructed out through the

Aid's serial port (for instance, when you send data to an external printer). The messages you have typed on the communication aid are stored as ASCII characters and sent out the serial port as ASCII characters.

Let's try that now. Simply construct a message and send it out the serial port with your communication aid or alternate access aid.

Example

Send out Serial Port	Result
this is a test	"this is a test" should appear on the computer screen

Typing Special Computer Keyboard Keys

In order to type out computer keys which are not part of the ASCII character set, you must use a special command called "press". To use it, you must send an <esc> (escape character) to the module, and then ",kbd,press," and then the keyname. Commands which consist of sequence of characters that follow the <esc> character will be referred to as "Escape Sequences".

Example

Send out Serial Port	Result
<esc>,kbd,press,home.	types the "home" key
<esc>,kbd,press,enter.	types the "enter" key
<esc>,kbd,press,f5.	types the "F5' key

You may put up to ten keynames after the press command.

Example

Send out Serial Port	Result
<esc>,kbd,press,pgup,home,enter.	types the "page up", "home" and "enter" key

You will find the complete list of keynames for the different computer keyboards in the reference section at the back of this manual. [NOTE: This section is not yet complete.]

Moving The Mouse

So far we have been only talking about the keyboard. In order to use the mouse, we must use the "mou" commands.

> NOTE: You must be running software that accepts the mouse as input. Otherwise, mouse commands will have no effect.

To move the mouse, we use the "move" command. The move command takes two numbers after it: first the motion in the horizontal direction and then the motion in the vertical direction. There must be either a "+" or "-"

sign before the number. Positive numbers move the mouse pointer to the right or down. Negative numbers move the mouse pointer left and up. The numbers can range from -64 to +63.

Example

Send out Serial Port	Result
<esc>,mou,move,+10,-20.	moves 10 pixels to right and 20 pixels up
<esc>,mou,move,-10,+20.	moves 10 pixels to left and 20 pixels down
<esc>,mou,move,+0,-10.	moves 0 pixels to right and 10 pixels up

It's a good idea to at least program some square to move the mouse in the four directions by one pixel, by 10, and by 25 pixels. This will allow you to make large, fast movements and small, fine movements.

Example

Send out Serial Port	Result
<esc>,mou,move,+1,+0.	moves 1 right
<esc>,mou,move,-1,+0.	moves 1 left
<esc>,mou,move,+0,+1.	moves 1 down
<esc>,mou,move,+0,-1.	moves 1 up
<esc>,mou,move,+5,+0.	moves 5 right
<esc>,mou,move,-5,+0.	moves 5 left
<esc>,mou,move,+0,+5.	moves 5 down
<esc>,mou,move,+0,-5.	moves 5 up
<esc>,mou,move,+25,+0.	moves 25 right
<esc>,mou,move,-25,+0.	moves 25 left
<esc>,mou,move,+0,+25.	moves 25 down
<esc>,mou,move,+0,-25.	moves 25 up

Clicking with the Mouse

In order to click the button on a mouse, you use the "click" command.

Example

Send out Serial Port	Result
<esc>,mou,click,left.	clicks the left button
<esc>,mou,click,right.	clicks the right button
<esc>,mou,click.	clicks the default button which is usually the left one

If there is no button mentioned in the command, then the T-TAM presses the default button. On multiple-button mice (e.g., the IBM mouse), the default button is the left button. On single-button mice (e.g., the Apple mouse), it is the only button.

Handling Errors

There are two types of errors which can be made. One is misspelling a word in an escape sequence command. If this type of error is made while using the T-TAM, a short low beep will be sounded. This means that the T-TAM has either ignored a misspelled word, or does not support a command and therefore is ignoring it.

The second type of error is more critical. It occurs when the rules of the GIDEI standard have been violated, and the T-TAM is unable to recover from the error. The T-TAM will continue to ignore all input until it receives the "acknowledge" command.

Example

Send out Serial Port	Result
`<esc>,ACK.`	clears the critical error

You must spell "ACK" in all upper case letters (see Intermediate Level for instructions on typing the shift key). Once the T-TAM receives the acknowledge command, it will return to being ready for ISO typing.

INTERMEDIATE LEVEL

Using Modifier Keys with Other Keys

Often you need to hold down a modifier key (e.g., shift, alt, ctrl, etc.) with another key. You can do this by using the "hold" command.

Example

Send out Serial Port	Result
<esc>,kbd,hold,shift.	hold the shift with next key(s)
<esc>,kbd,press,f5.	types a shift-f5

Example

Send out Serial Port	Result
<esc>,kbd,hold,ctrl,alt.	hold the ctrl and alt keys with the next key(s)
<esc>,kbd,press,kpdel.	types a ctrl-alt-del, which reboots an IBM computer.

The hold command lets you program a single square on your communication or alternate access aid to let you hold down modifier keys with other keys. You can hold up to ten keys at a time.

Typing More than One Key at at time

There may be some situations where you need to have the computer think you are holding a number of keys down at the same time. For example, you may want to program the reboot keystrokes into one square on you communication aid. You can do this as follows by using the combine command.

Example

Send out Serial Port	Result
<esc>,kbd,combine,ctrl,alt,kpdel.	types the ctrl-alt-del key sequence.

You can combine up to ten keys at a time.

Dragging with the Mouse

In order to perform dragging function with the mouse, you can program a square on your Aid to perform the drag, or to use a combination of the "lock" & "release" commands with the "move" command.

Example

Send out Serial Port	Result
`<esc>,mou,lock,left.`	pushes the left button down
`<esc>,mou,release,left.`	releases the left button
`<esc>,mou,lock.`	pushes the default button which is usually the left one

A drag is a combination of pushing a button down, moving the mouse, and then releasing the button. To do this, you would use the three commands like this:

Example

Send out Serial Port	Result
`<esc>,mou,lock,left.`	pushes the left button
`<esc>,mou,move,+20,+30.`	move 20 pixels to the right and 30 pixels down.
`<esc>,mou,release,left.`	releases the left button

If you are programming squares on your aid, it would be wise to program one to lock a button down and another one to release a button, and to program several buttons so you can move the mouse around. Therefore, to perform a drag, you will need to make at least three selections.

ADVANCED LEVEL

In this section, we will discuss advanced commands. These commands are generally not needed except for advanced users.

Locking and Releasing Keys

If you want to have complete control over how keys are typed, you can use the "lock" and "release" commands. The lock command presses the key down, and the release command lets the key up.

Example

Send out Serial Port	Result
`<esc>,kbd,lock,f1.`	press f1 down
`<esc>,kbd,lock,f2.`	press f2 down
`<esc>,kbd,release,f1.`	let f1 up
`<esc>,kbd,lock,f3.`	press f3 down
`<esc>,kbd,release,f2.`	let f2 up
`<esc>,kbd,release,f3.`	let f3 up

There is usually no need to use the lock and release command since most typing functions can be accomplished with the "press", "hold", and "combine" commands mentioned in the Intermediate Level section.

Changing the Baud Rate of the Serial Port

If your Aid can communicate at a faster rate than 300 baud, you can use the "baudrate" command to change the baud rate to the higher baud rate which you device supports.

Example

Send out Serial Port	Result
`<esc>,comm,baudrate,9600.`	change baud rate to 9600 baud
`<esc>,comm,baudrate,300.`	change baud rate back to 300 baud

GENERAL REFERENCE

Basic Structure of Commands

1) For typing ASCII characters: Just send the ASCII character.

2) For other keyboard and mouse operations: Send an "escape" character, followed by the name of the sub-standard being used, followed by the commands and the parameters. All sequences must begin with an escape, end with a period, and have commas between the items. Upper or lower case letters can be used. A generic command looks like this:

```
<esc>,sub-standard name,command,[parameter1,parameter2].
```

There may be zero, one, two or more parameters in a command.

Sub-Standard Names

There are four sub-standard names that can be specified in commands. These are:

In command	Meaning
kbd	keyboard
mou	mouse
comm	communication
gen	general

Command Names

NOTE: The commands are listed below, followed "generic" command line and an explanation of what the command does. These generic command lines sometimes substitute generic terms for the what you would actually type: for instance if you see "keyname1," that just means "any key name you want to use." Square brackets [] indicate an item that is optional: it may or may not be in a particular command line.

1) Keyboard Commands

Below is a summary of all the keyboard commands. These commands begin with an escape, followed the designation of the keyboard standard:

```
<esc>,kbd,command,[parameter1,parameter2].
```

In these commands, keynames are used as parameters. There may be zero, one, two or more parameters.

Lock Command

```
<esc>,kbd,lock,keyname1, . . . [keyname10].
```

Presses the key(s) down and keeps it down until released with the "release" command. There must be at least one keyname after the lock command. Up to ten keynames can be included after the command.

Release Command

```
<esc>,kbd,release,[keyname1] . . . [keyname10].
```

Releases the key(s) up. Up to ten keynames may follow the command. If no keyname follows the command, all "locked" keys will be released.

Press Command

```
<esc>,kbd,press,keyname1, . . . [keyname10].
```

Presses the key(s) down and then releases them in the same order as the keynames. There must be at least one keyname after the lock command. Up to ten keynames can be included after the command.

Combine Command

```
<esc>,kbd,combine,keyname1, . . . [keyname10].
```

Presses the key(s) down in the same order as the keynames and then releases the key(s) in the reverse order as the keynames. There must be at least one keyname after the combine command. Up to ten keynames can be included after the command.

Hold Command

```
<esc>,kbd,,hold,keyname1, . . . [keyname10].
```

This command does not get executed until the next set of keys are pressed. This command holds the key(s) down in the same order as the keynames, then "presses" or "combines" the next set of keys, then releases the key(s) in the reverse order as the keynames. There must be at least one keyname after the hold command. Up to ten keynames can be included after the command.

2) Mouse Commands

Below is a summary of all the mouse commands. These commands begin with an escape, followed the designation of the mouse standard:

```
<esc>,mou,command,[parameter1,parameter2].
```

In these commands, positions or buttons are used as parameters. There may be zero, one, two or more parameters, depending on the command and what you want to use it to do.

Move Command

```
<esc>,mou,move,horizontal distance,vertical distance.
```

This command moves the mouse a specified distance. The command should be followed by two numbers: first the motion in the horizontal direction and then the motion in the vertical direction. There must be either a "+" or "-" sign before the number. Positive numbers move the mouse pointer to the right or down. Negative numbers move the mouse pointer left and up. The numbers can range from -64 to +63.

Goto Command

```
<esc>,mou,goto,horizontal location,vertical location.
```

This command moves the mouse pointer to a particular location on the screen. The location is specified by a pair of numbers which follow the Goto command. "Goto" must be written as one word (no space in between).

Click Command

```
<esc>,mou,click,[button,button].
```

This command presses the mouse button down and then releases it right away; in other words, it's just like clicking once on the mouse button. If there is no button mentioned in the command, then the T-TAM presses the default button. On multiple-button mice (e.g., the IBM mouse), the default button is the left button. On single-button mice (e.g., the Apple mouse), it is the only button.

Lock Command

```
<esc>,mou,lock,[button,button].
```

This command presses the mouse button down and holds it down until you send a "release" command. The buttons are designated in the same way as for the "click" command. "Dragging" operations with the mouse can be executed using the "lock," "move" and "release" commands, in that order.

Release Command

```
<esc>,mou,release,[button,button].
```

This command releases the mouse button when it is being held down by the "lock" command. The buttons are designated in the same way as for the "Click" command. "Dragging" operations with the mouse can be executed using the "lock," "move" and "release" commands, in that order.

3) *Communication Commands*

Baudrate Command

If your Aid can communicate at a faster rate than 300 baud, you can use the "baudrate" command to change the T-TAM's baud rate to the higher baud rate which your device supports.

```
<esc>,comm,baudrate,new rate.
```

4) *General Commands*

Version Command

[This command is not included in this version of the T-TAM Manual.]

Reset Command

[This command is not included in this version of the T-TAM Manual.]

Acknowledge Command

[See the "Intermediate" section of this manual.]

TECHNICAL REFERENCE

Introduction

This section contains a detailed description of every command of the GIDEI standard.

The overall GIDEI standard is composed of multiple sub-standards. At the present time, the four sub-standards are:

- Keyboard (kbd)
- Mouse (mou)
- Communication (comm)
- General (gen)

We will first describe the general GIDEI standard, and then the sub-standards one at a time.

The Standard Based on Single-Byte Numbers

So far, we have described the commands by use of what we will call "aliases". That is, the actual GIDEI commands are based on single-byte numbers or what we will refer to as "code." Since some devices cannot produce all of these numbers, and since in some cases a user must program the Aid, we allow English-like aliases to be used in place of these single-byte numbers.

GIDEI standard

The basic structure of the GIDEI standard is an escape character (<esc>) followed by the first sub-standard that you want to use, and then the command and parameters of that command.

The exact structure after indicating the sub-standard will vary, but the general structure is:

General Structure

```
<esc><standard><command><parameters> . . . [<parameters>]
```

The different standards for this implementation of GIDEI are:

Standards

Code	Aliases	Meanings
<1>	kbd	keyboard std
<2>	mou	mouse std
<100>	comm	communication std
<101>	gen	general std

BEGIN and END Command

Syntax: <begin>

Code: <254>

Aliases: BEGIN

Syntax: <end>

Code: <253>

Aliases: END

The "BEGIN" command, combined with the "END" command allows you to be more efficient with doing several commands together. If you need to do several keyboard and mouse commands, you can group them together with a begin and end.

Example

```
Send out Serial Port              Result
<esc>,BEGIN,                      start grouping
  kbd,lock,shift.                 lock down shift key
  mou,click,left.                 click left button on mouse
  kbd,rel,shift.                  release shift
END.                              end grouping
```

You can also group commands within a standard.

Example

```
Send out Serial Port              Result
<esc>,kbd,BEGIN,                  start grouping
  lock,ctrl.                      lock down ctrl key
  press,f1,f2,home.               press f1, f2 and home keys
  rel,ctrl.                       release ctrl
  press,esc.                      press escape key
END.                              end grouping
```

If you want, you can even do several level of grouping.

Example

```
Send out Serial Port              Result
<esc>,BEGIN,                      start grouping
  kbd,lock,shift.                 lock down shift key
  kbd,BEGIN,                      start second grouping
    lock,ctrl.                    lock down ctrl key
    press,f1,f2,home.             press f1, f2 and home keys
    rel,ctrl.                     release ctrl
    END.                                   end second grouping
    mou,click,left.                        click left button on mouse
    kbd,rel,shift.                         release shift
  END.                                     end grouping
```

Again, you do not need to use these commands. They are there to let you simplify the commands when doing a series together.

Terminate Command

```
Syntax:    <terminate>
Code:      <252>
Aliases:   .
```

You have been using the terminate command without even realizing it. After each command, we have been ending the command with a period. The period is the alias for the terminate byte.

Acknowledge Command

```
Syntax:    <acknowledge>
Code:      <251>
Aliases:   ACK
```

There are two levels of errors. One that is signalled by a short low beep. This means that an command or standard was given which was not understood and therefore skipped. When a more serious error occurs, a long high beep is given. This means that the command was stopped because of a catastrophic error was detected, and all input will be ignored until an acknowledge command is received. The acknowledge command will then clear the error and return back to ISO typing.

Example

Send out Serial Port	Result
<esc>,ACK.	Clear Error Condition

Keyboard Commands

Below is a summary of all the keyboard commands. These commands follow the designation of the keyboard standard:

```
Syntax:    <esc><kbd><command>[<parameter> . . . <parameter>]
Code:      <1>
Aliases:   kbd
```

In these commands, keynames are used as parameters. Keynames for the different keyboards are listed in an Appendix. [NOTE: That appendix is not included with this beta version of the documentation.]

Lock Command

```
Syntax:    <lock><keyname1> . . . [<keyname10>]

Code:      <1>

Aliases:   lock
```

Presses the key(s) down and keeps it down until released with the "release" command. There must be at least one keyname after the lock command. Up to ten keynames can be included after the command. The BEGIN command can be included immediately after the lock command in order to do multiple groups of keynames beyond the 10 limit. The BEGIN command can not occur after the first keyname.

Release Command

```
Syntax:    <release>[<keyname1>] . . . [<keyname10>]

Code:      <2>

Aliases:   release
           rel
```

Releases the key(s) up. Up to ten keynames may follow the command. If no keyname follows the command, all "locked" keys will be released. The BEGIN command can be included immediately after the release command in order to do multiple groups of keynames beyond the 10 limit. The BEGIN command can not occur after the first keyname.

Press Command

```
Syntax:    <press><keyname1> . . . [<keyname10>]

Code:      <3>

Aliases:   press
```

Presses the key(s) down and then releases them in the same order as the keynames. There must be at least one keyname after the lock command. Up to ten keynames can be included after the command. The BEGIN command can be included immediately after the press command in order to do multiple groups of keynames beyond the 10 limit. The BEGIN command can not occur after the first keyname.

Combine Command

```
Syntax:    <combine><keyname1> . . . [<keyname10>]
Code:      <4>
Aliases:   combine
```

Presses the key(s) down in the same order as the keynames and then releases the key(s) in the reverse order as the keynames. There must be at least one keyname after the combine command. Up to ten keynames can be included after the command. The BEGIN command can be included immediately after the combine command in order to do multiple groups of keynames beyond the 10 limit. The BEGIN command can not occur after the first keyname.

Hold Command

```
Syntax:    <hold><keyname1> . . . [<keyname10>]
Code:      <5>
Aliases:   hold
```

This command does not get executed until the next set of keys are pressed. This command holds the key(s) down in the same order as the keynames. Then "presses" or "combines" the next set of keys. Then releases the key(s) in the reverse order as the keynames. There must be at least one keyname after the hold command. Up to ten keynames can be included after the command. The BEGIN command can be included immediately after the hold command in order to do multiple groups of keynames beyond the 10 limit. The BEGIN command can not occur after the first keyname.

[NOTE: The "Technical Reference" section is not complete beyond this point.]

Mouse Commands

Move Command

```
Syntax:    <move>
Code:      <4>
Aliases:
```

Goto Command

Syntax:	<goto>
Code:	<5>
Aliases:	

Click Command

Syntax:	<click>[<button>] . . . [<button>]
Code:	<3>
Aliases:	click

Lock Command

Syntax:	<lock>[<button>] . . . [<button>]
Code:	<1>
Aliases:	lock

Release Command

Syntax:	<release>[<button>] . . . [<button>]
Code:	<2>
Aliases:	release rel

Communication Commands

Baudrate Command

Syntax:	<>
Code:	<>
Aliases:	

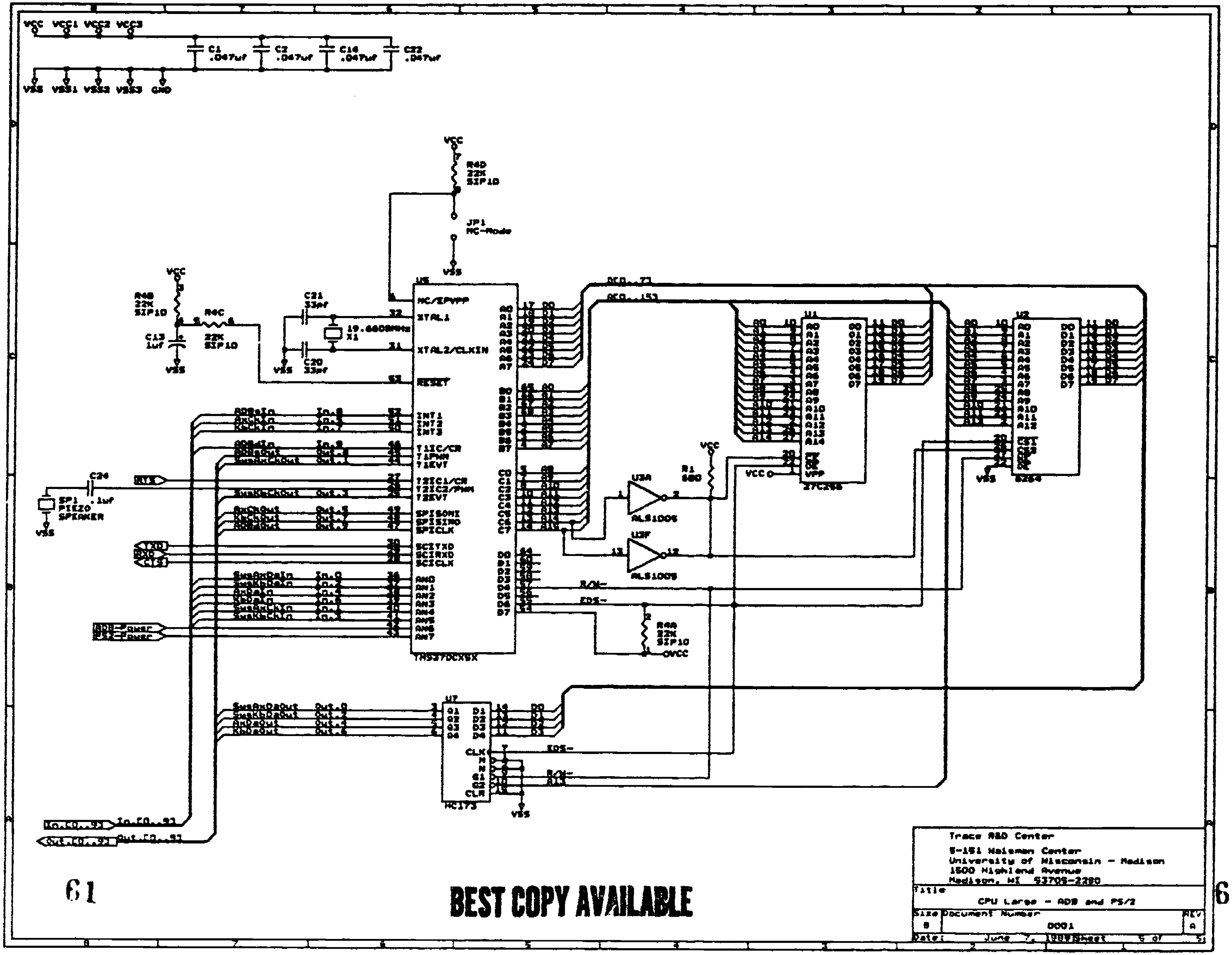
General Commands

Version Command

Syntax: <>
Code: <>
Aliases:

Reset Command

Syntax: <>
Code: <>
Aliases:

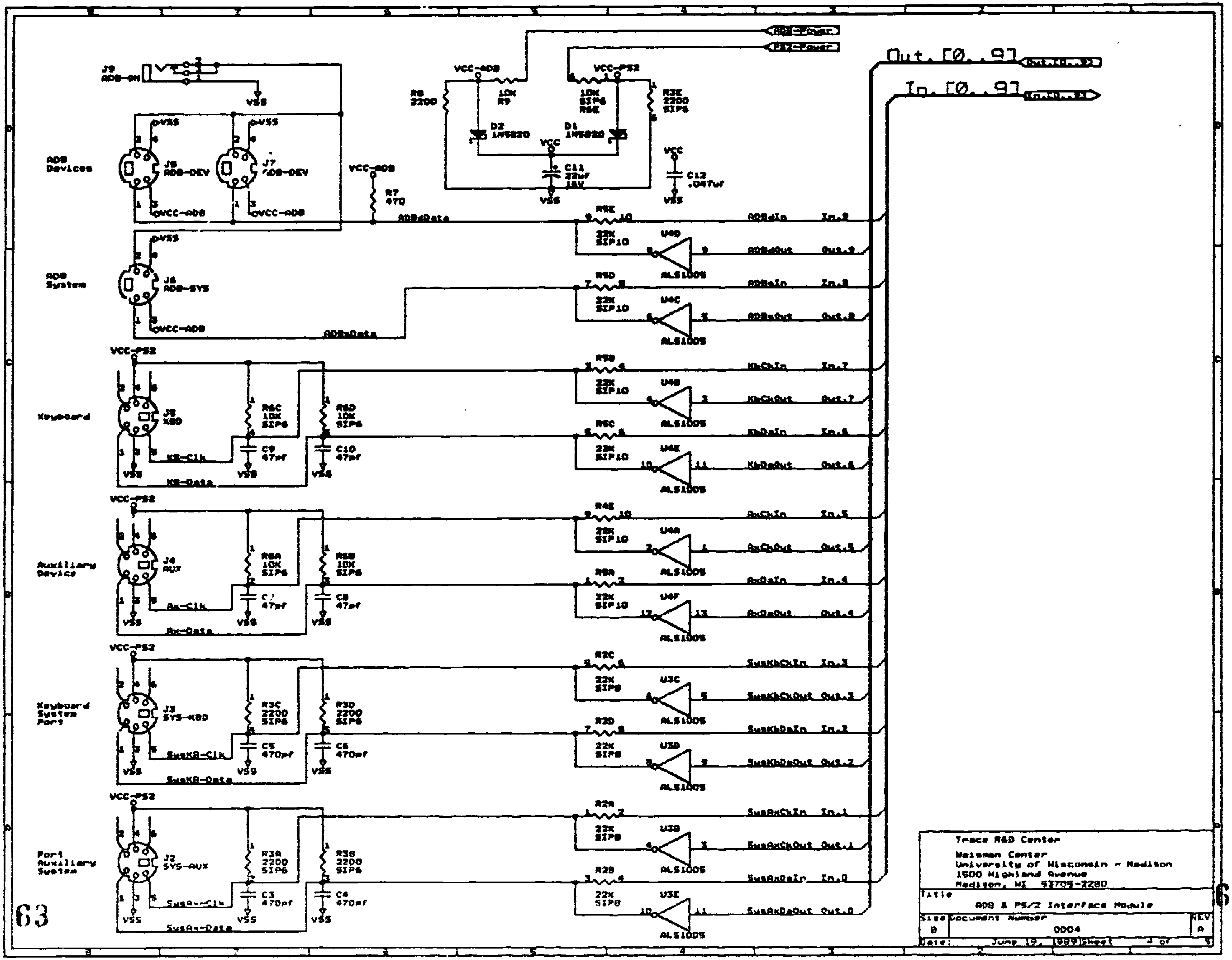


Trace R&D Center
S-151 Waisman Center
University of Wisconsin - Madison
1500 Highland Avenue
Madison, WI 53705-2280

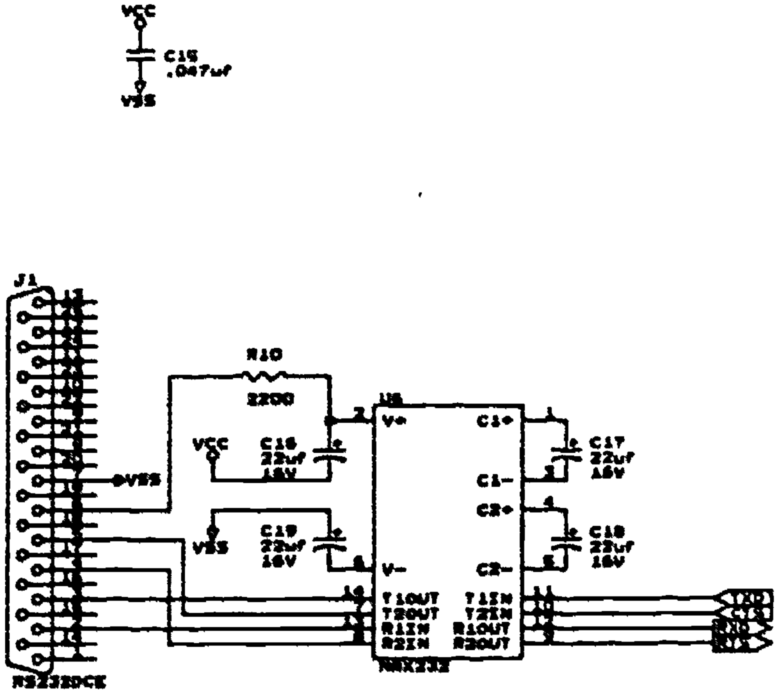
Title
CPU Large - ADB and PS/2

Size	Document Number	REV
B	0001	A

Date: June 7, 1989 Sheet 5 of 5



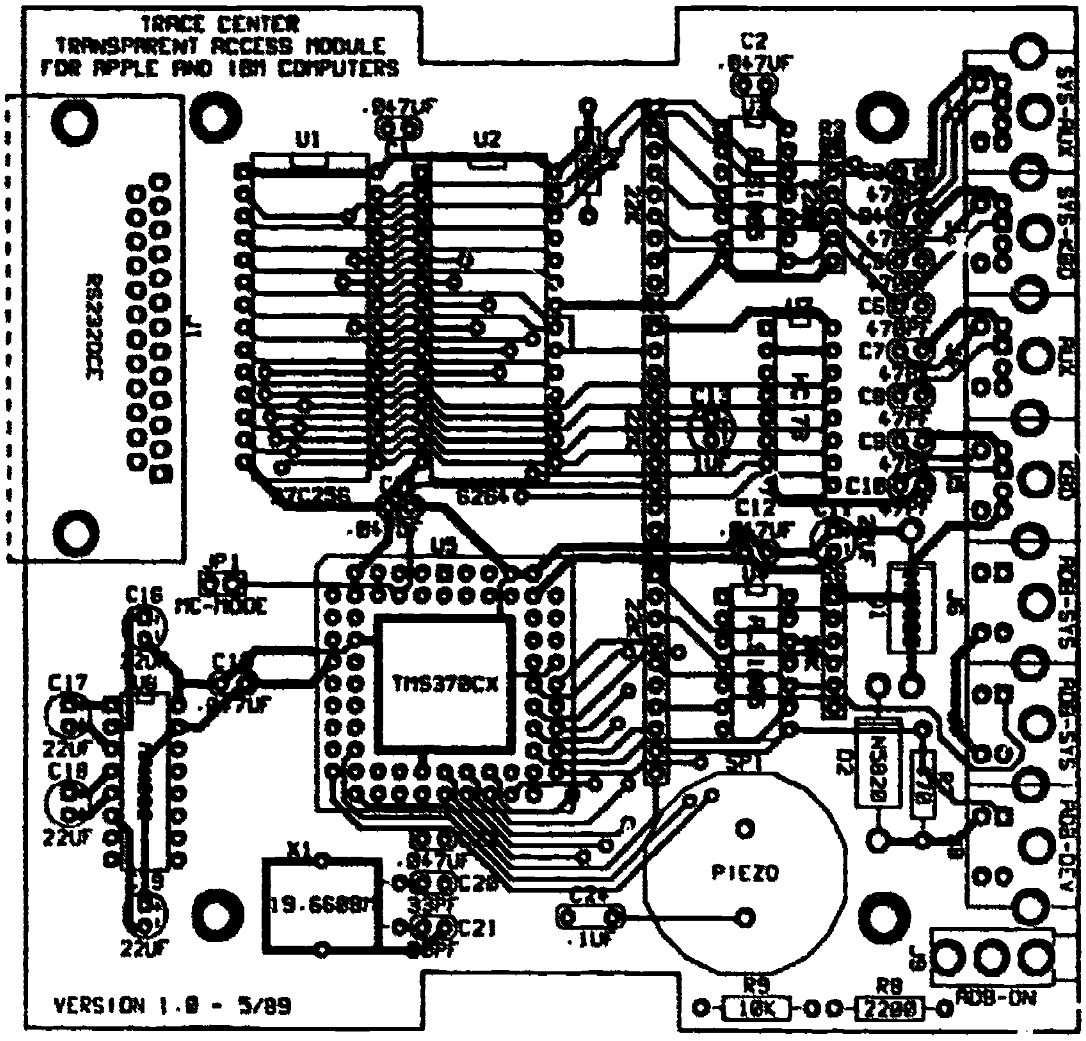
J9
ADB-ON
VSS
ADB Devices
J8
ADB-DEV
J7
ADB-DEV
VCC-ADB
ADB System
J6
ADB-SYS
ADBsData
VCC-ADB
R7
470
ADBdData
VCC-ADB
R8
2200
10K
R9
D2
1N5820
VCC
C11
22uf
16V
VSS
VCC-PS2
10K
SIP6
R6E
D1
1N5820
R3E
2200
SIP6
VCC
C12
.047uf
VSS
ADB-Power
PS2-Power
Out.[0..9]
Out.[0..9]
In.[0..9]
In.[0..9]
R5E
22K
SIP10
U4D
ALS1005
ADBdIn
In.9
ADBdOut
Out.9
R5D
22K
SIP10
U4C
ALS1005
ADBsIn
In.8
ADBsOut
Out.8
Keyboard
VCC-PS2
J5
KBD
R6C
10K
SIP6
R6D
10K
SIP6
C9
47pf
C10
47pf
KB-Clk
KB-Data
VSS
R5B
22K
SIP10
U4B
ALS1005
KbClkIn
In.7
KbClkOut
Out.7
R5C
22K
SIP10
U4E
ALS1005
KbDaIn
In.6
KbDaOut
Out.6
Auxiliary Device
VCC-PS2
J4
AUX
R6A
10K
SIP6
R6B
10K
SIP6
C7
47pf
C8
47pf
Ax-Clk
Ax-Data
VSS
R4E
22K
SIP10
U4A
ALS1005
AxClkIn
In.5
AxClkOut
Out.5
R5A
22K
SIP10
U4F
ALS1005
AxDaIn
In.4
AxDaOut
Out.4
Keyboard System Port
VCC-PS2
J3
SYS-KBD
R3C
2200
SIP6
R3D
2200
SIP6
C5
470pf
C6
470pf
SysKB-Clk
SysKB-Data
VSS
R2C
22K
SIP8
U3C
ALS1005
SysKbClkIn
In.3
SysKbClkOut
Out.3
R2D
22K
SIP8
U3D
ALS1005
SysKbDaIn
In.2
SysKbDaOut
Out.2
Port Auxiliary System
VCC-PS2
J2
SYS-AUX
R3A
2200
SIP6
R3B
2200
SIP6
C3
470pf
C4
470pf
SysAx-Clk
SysAx-Data
VSS
R2A
22K
SIP8
U3B
ALS1005
SysAxClkIn
In.1
SysAxClkOut
Out.1
R2B
22K
SIP8
U3E
ALS1005
SysAxDaIn
In.0
SysAxDaOut
Out.0
Trace R&D Center
Waisman Center
University of Wisconsin - Madison
1500 Highland Avenue
Madison, WI 53705-2280
Title
ADB & PS/2 Interface Module
Size
B
Document Number
0004
REV
A
Date:
June 19, 1989
Sheet
4 of 5



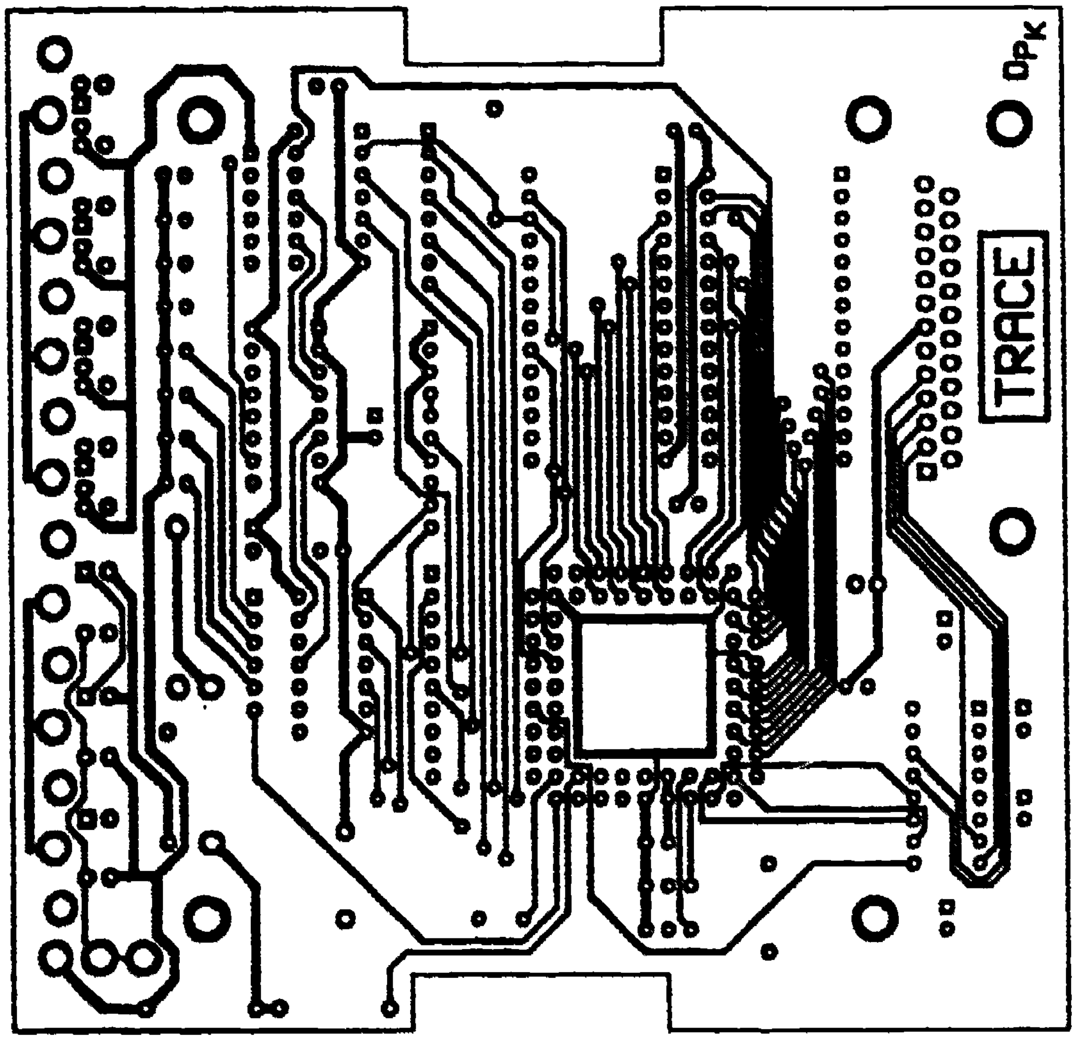
Trace R&D Center

Waisman Center
University of Wisconsin - Madison
1500 Highland Avenue
Madison, WI 53705-2280

Title		
EIA Standard RS-232-C DCE Module		
Size	Document Number	REV
B	1004	A
Date: August 28, 1989	Sheet 4 of 4	



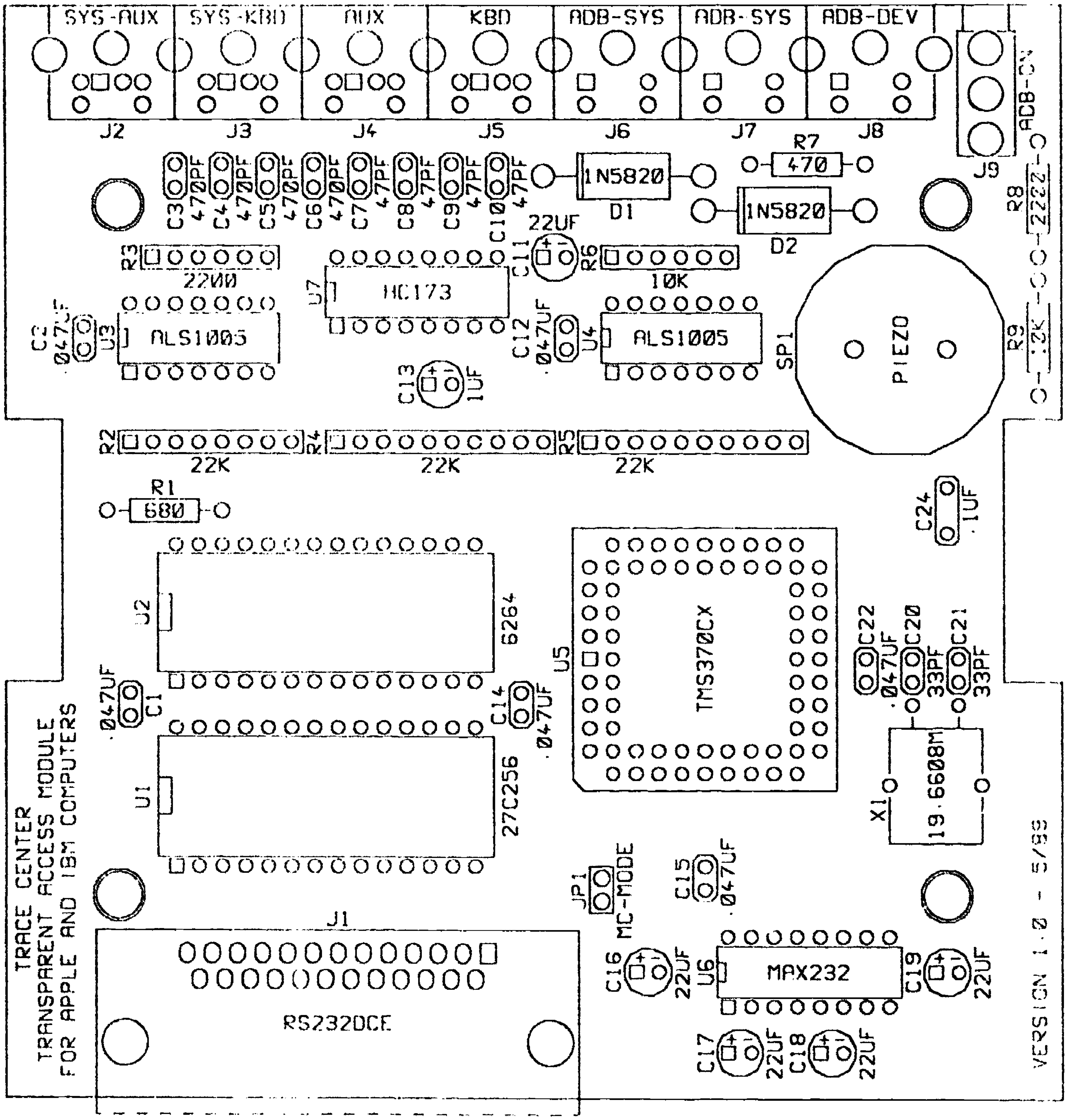
Component Side View



Solder Side View

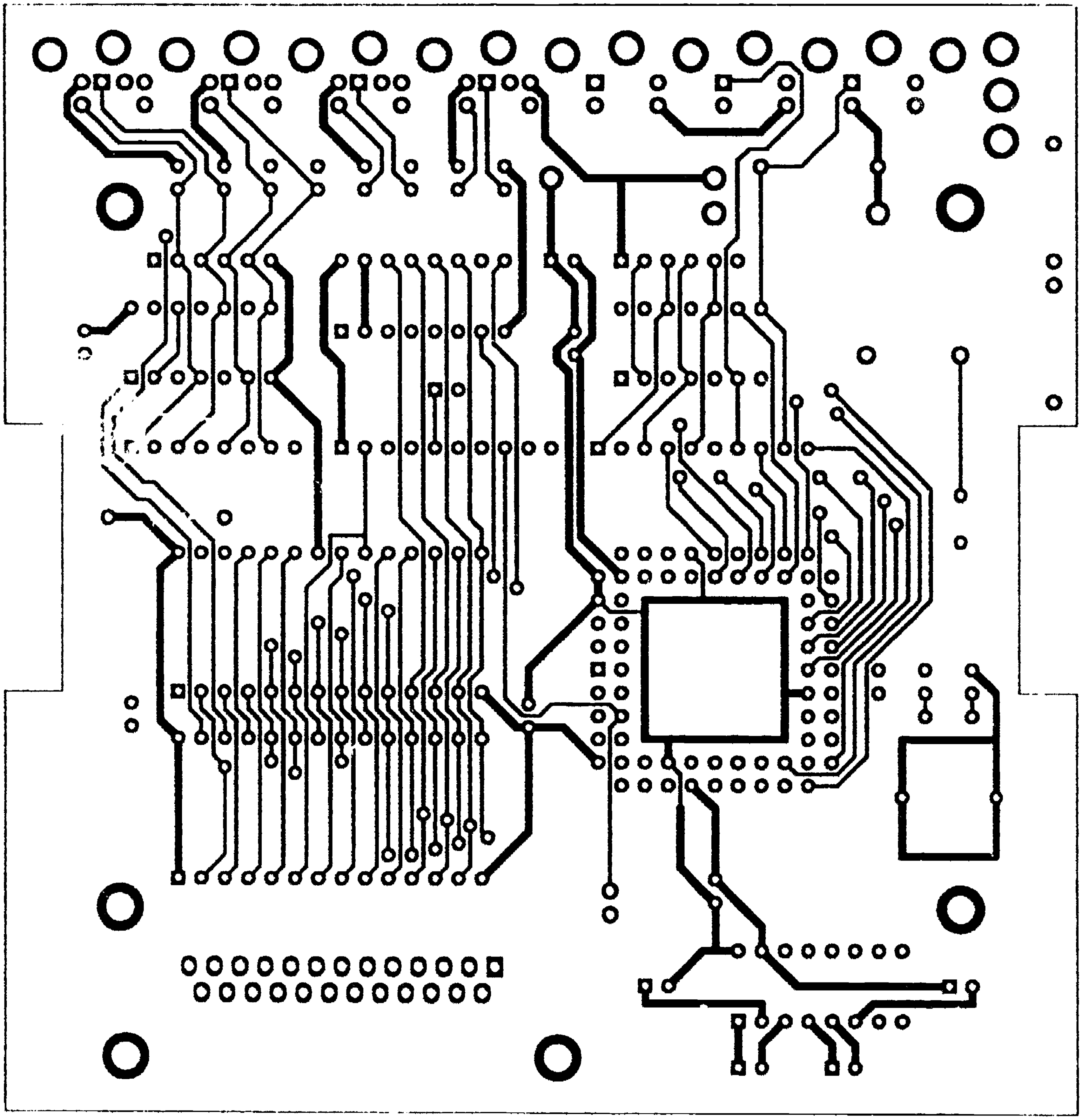
T-TAM PC Board Version 1.0

Silk-screen Layer - Component Side
(Top View)



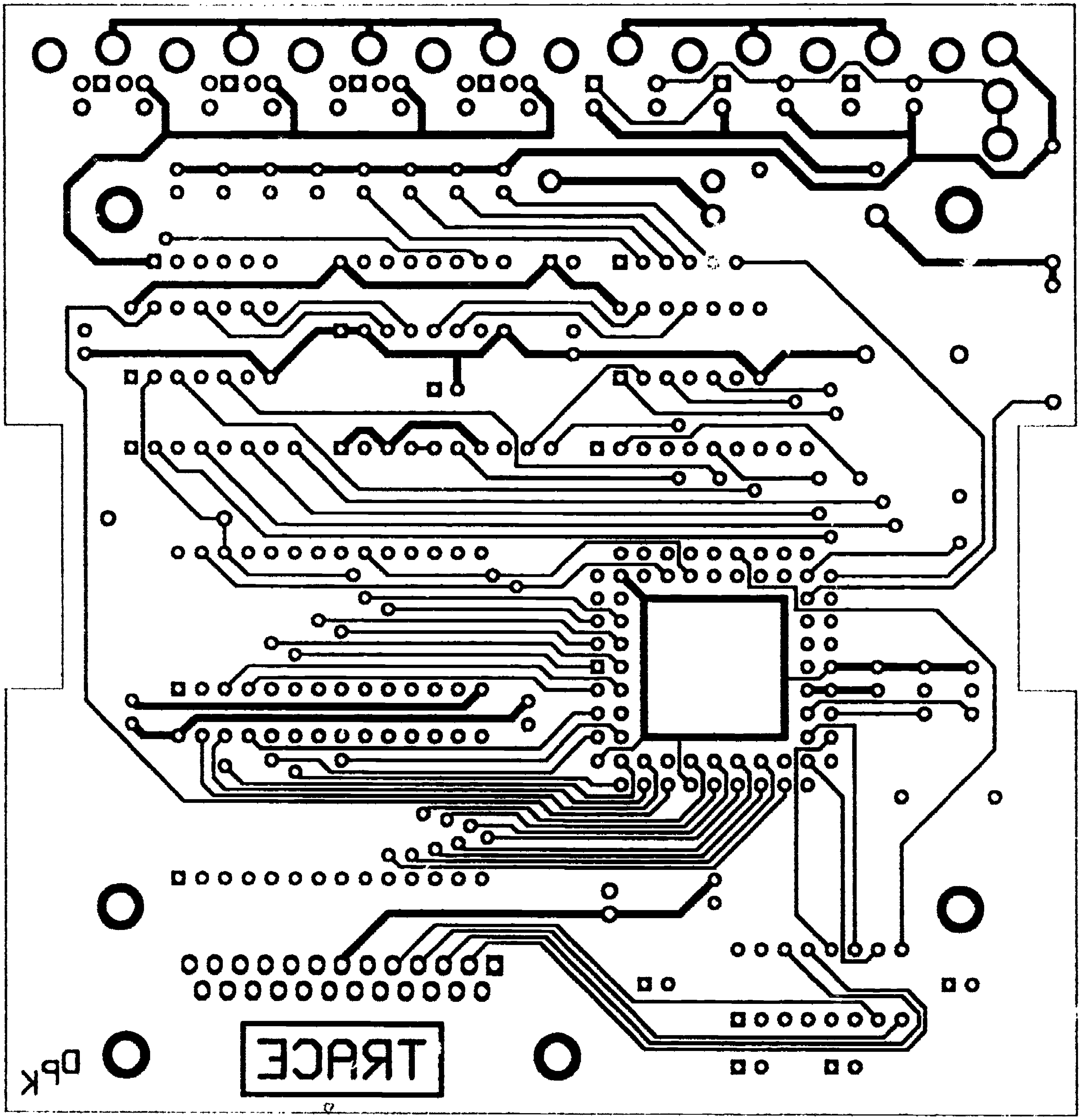
T-TAM PC Board Version 1.0

Copper Layer - Component Side
(Top View)



T-TAM PC Board Version 1.0

Copper Layer - Solder Side
(Top View)



T-TAM PC Board Version 1.0

Software is available upon request. Please send requests to:

T-TAM Project Manager
Trace Research and Development Center
University of WI - Madison
1500 Highland Ave.
Madison, WI 53705

(608) 262-6966