CHAPTER 19
UNIVERSITY OF WASHINGTON

**Departments of Electrical Engineering and
Computer Science & Engineering
Seattle, Washington 98195**

Principal Investigators:

Yongmin Kim (206) 685-2271
Gaetano Borriello (206) 685-9432

Programmable Timer Implemented in Surface Mount Technology

Designer: Trond Arild Lode Tobiassen
Client Coordinator: Robert Nansel
Fircrest School, Seattle, WA
Supervising Professor: Dr. Yongmin Kim
Department of Electrical Engineering
University of Washington
Seattle, WA 98105

INTRODUCTION

This programmable timer is to be used in a number of applications. As with all Fircrest projects, it is to be used in the training programs of disabled students. In particular, the circuit could be used to stimulate interest in an activity such as TV-watching. When the circuit is triggered, the timing period starts and a switch turns on the TV, and it is turned off when the timing period is over. During time-out, the start of another timing period is inhibited by the circuit. It is implemented using surface mount components to minimize size. This way, it can fit into the casing of other circuits.

SUMMARY OF IMPACT

The circuit can be used in any application where there is a need for turning equipment on or off for a period of time, from around four seconds to eight minutes and fifty seconds. Its current handling capability provides opportunity for the switching of relays and other applications where relatively high currents are involved. Its small size and low power consumption, in addition to having a separate power supply, make it even more attractive in these applications.

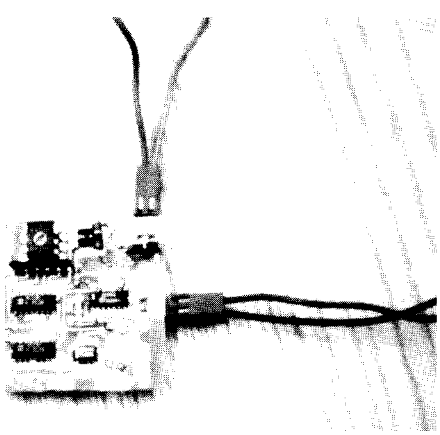


Figure 19.1. A Versatile Timer Module.

TECHNICAL DESCRIPTION

The circuit is made of surface mount CMOS, and MOSFET components to minimize size and increase battery lifetime. It is powered by a 3V lithium coin cell. The basic time controller is a TLC555D that is the classic 555 timer. It is resistor/capacitor-controlled and gives a clock pulse to a 74HC191 4-bit, binary, presettable, up/down counter that operates as a prescaler to increase the operating clock frequency of the 555, thus providing better stability. Its output is fed to another 191 whose presettable inputs are controlled by a 4-bit rotary switch. This rotary switch is not surface mounted, but the pinout and pad size are equivalent to one that is available. The 191's output goes low at the end of the time period and sets a NAND-gate flip-flop whereas triggering is enabled through the reset on the flip-flop.

The trigger is inverted twice to buffer slow or inadequate transitions. The output of the first inverter provides an inverted trigger and can be jumpered to the flip-flop input if an active-high trigger is preferred.

The outputs of the flip-flop have pull-down resistors to be able to saturate the MOSFET. Either one can be chosen by means of jumpers. Also, the flip-flop controls the timer by asserting a low on its reset pin when the timing period is over. The resistor-capacitor network on the 555 consists of a 0.33 μF capacitor, a 1 $M\Omega$ and a O-to-4 $M\Omega$ resistors. This, combined with the presettable counter (1 to 16 different periods) provides a range of about 4 sec. to 8 minutes and fifty sec. In the test circuit, 20 sec. to 5 minutes and 20 sec. is selectable in multiples of 20 sec. The MOSFET currently used is not the originally intended surface mount component because that one proved to have some difficulty being turned on. Thus, the one used under testing has been applied, although it has high on-resistance and would probably burn if the load current took on the values that were specified originally (1 Amp.). However, there are several other MOSFETs available in the same packaging and it could be part of a future project to find one that has low on-resistance and gets turned on by 2-3V. The cost of the circuit is approximately $10.

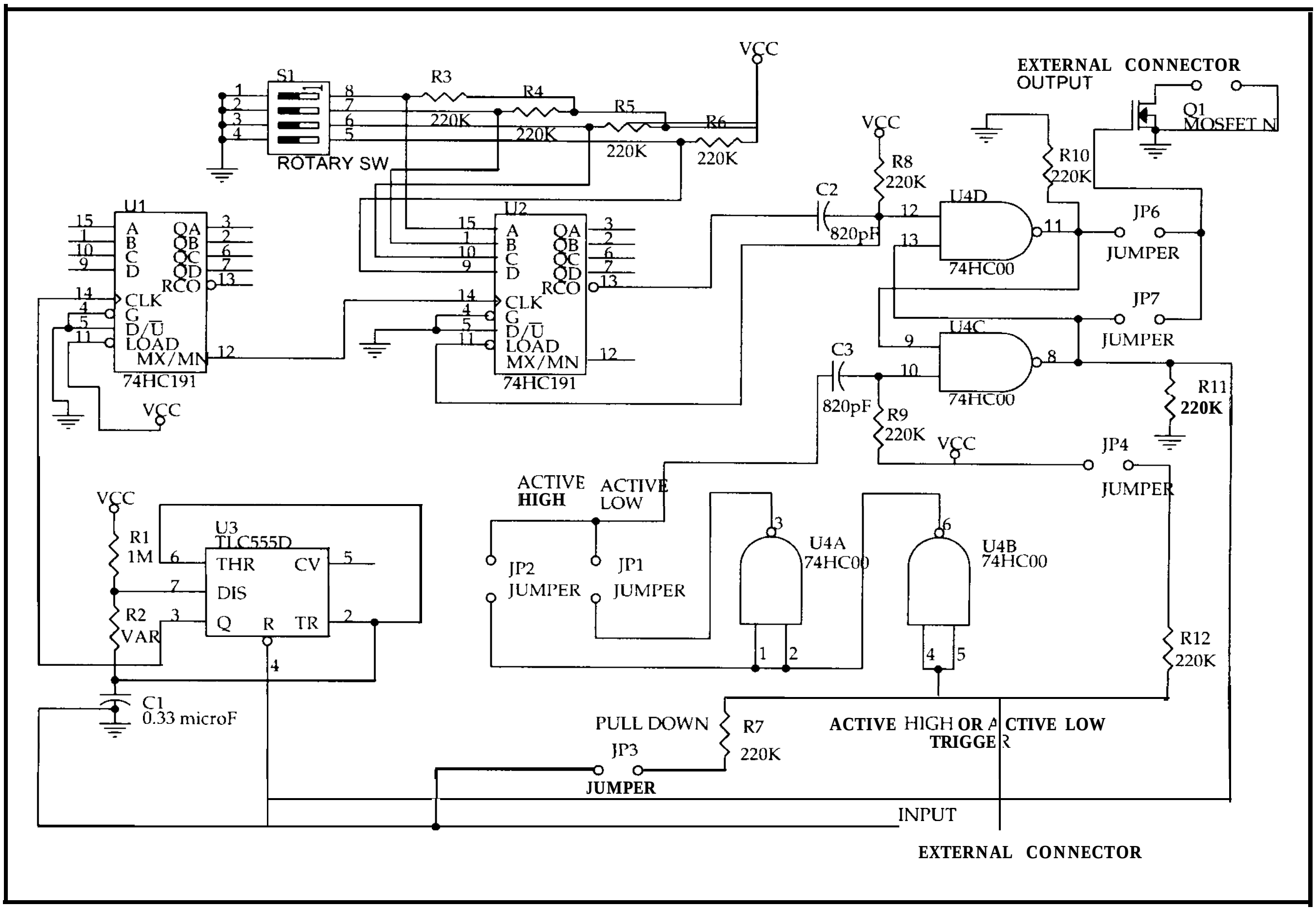


Figure 19.2. Circuit Diagram of Programmable Timer.

A Device for Training the Spinal Cord Injured Patient and Monitoring their Activities

Designers: Aaron Ching, Yuhong Xiong
Client Coordinator: Robert Nansel
Fircresf School, Seattle, WA
Supervising Professor: Dr. Yongrnin Kiln
Department of Electrical Engineering
University of Washington
Seattle, WA 98195

INTRODUCTION

Pressure sores are one of the most frequent and serious complications affecting spinal cord-injured individuals newly confined to wheelchairs. While many factors contribute to the occurrence of the pressure sore problems, in almost all cases the major cause is the excessive pressure being applied to the lower part of body over an excessive duration of time. It is therefore recommended that patients newly confined to a wheelchair do a “push-up” of their bodies at regularly timed intervals.

In order to aid these individuals, the Pressure Relief Reminder (PRR) is necessary to train them to form the unconscious habit of doing regular push-ups and to monitor their compliance with the prescribed push-up exercises. The device is suitable for institutional use as well as for other users whose mobility is impaired and are confined to wheelchair or bed.

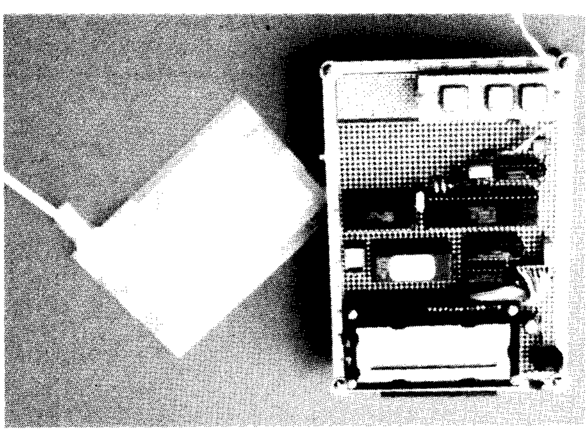


Figure 19.3. Pressure Relief Reminder.

SUMMARY OF IMPACT

The Pressure Relief Reminder helps therapists to train patients in order to form the unconscious habit of doing regular push-ups. According to statistics, there are about 177,000 people in the U.S. experiencing spinal cord-related injury in 1991. Many of them need to spend additional resources for further medical treatment as a result of forgetting to do the pressure relief exercises. This device not only saves avoidable medical expenses, but also facilitates therapists in adjusting the recovery process with respect to different needs. In addition, a memory system is implemented to record the pressure relief routine of the patients. Therapists can provide more detailed and accurate recommendations based on the readout from the memory.

TECHNICAL DESCRIPTION

The PRR is a 3-button battery-powered device that is mounted on the back of a wheelchair. A force sensor is extended from the device and is placed under the cushion of the wheelchair. If a patient sits in the wheelchair for a longer time than the programmed value, an audio or visual alarm will continuously signal the patient until a valid push-up has been done. Otherwise, the device will not be noticed if the patient does the prescribed exercise within the expected time period. During the whole process, records about the time of day, actual relief time, and inter-relief time are stored in the memory for future reference.

When used by the therapists, the PRR can be programmed to meet the specific needs of each patient. Three parameters can be entered by using the buttons on the device: relief time, inter-relief time, and a real-time clock. The pressure threshold of the force sensor can be adjusted by tuning a turn trimmer. Also, therapists can read the records from the LCD to evaluate the activities of the patient. The PRR has three major functions: monitoring the patient's activities, programming different parameters, and storing and displaying the records.

The hardware of the PRR consists of four major blocks: control, memory, user interface, and power. The control block consists of the Intel 80C31 micro-controller. HM6116 SRAM is used as the data memory, whereas 27C64 EPROM is used to store the instruction code for the microcontroller. The user interface consists of a piezo alarm, an LED, an LCD, three push buttons and a pressure sensor. The piezo alarm and the LED provide the patient with audio and visual alarms. The LCD is used to display the various parameters and records in the memory. An Interlink force sensing resistor is used as the pressure transducer. It changes the resistance according to the magnitude of force applied to it.

Four AA-size rechargeable batteries are used as the power source. Circuitry is built to recharge the batteries. Due to the limited battery capacity, the patient must assume the responsibility of recharging the device every night. The estimated cost to manufacture the PRR is $40.

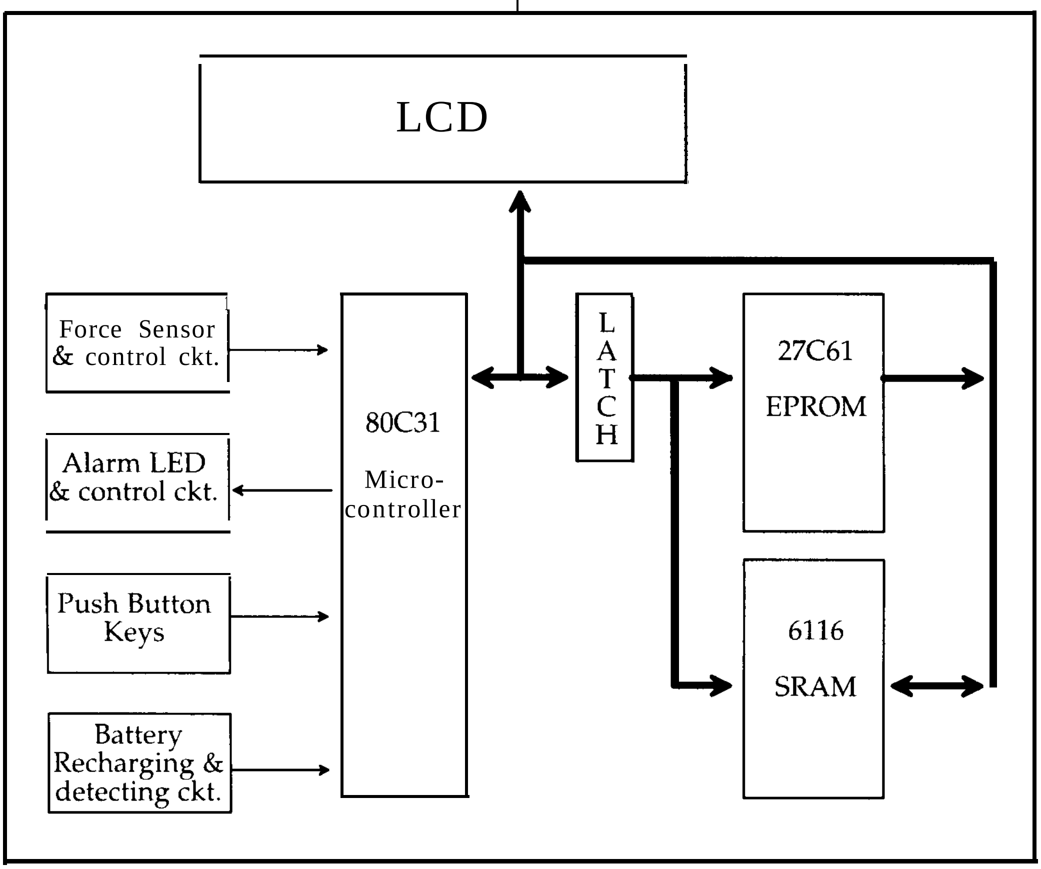


Figure 19.4. Functional Block Diagram of the PRR Device.

A Device to Assist the Physically Handicapped in Everyday Activities

Designers: Keith Kilmer, Crandall Chow
Client Coordinator: Dr. John Eiler
Fircresf School, Seattle, WA
Supervising Professor: Dr. Yongmin Kim
Department of Electrical Engineering
University of Washington
Seattle, Washington 98195

INTRODUCTION

The X-10 Remote Control System is a product already on the market that uses a palm-sized RF remote control for household items. This remote control incorporates 16 individual buttons to turn lights and appliances on or off. The 16-button remote works well for the physically able, but is difficult for some handicapped people since close arrangement of buttons would require rather precise motor control that may be lacking.

The X-10 Remote Controller Modification does away with the need for 16 individual buttons, providing complete functionality with only one button. It interfaces with the original remote control and uses an array of 16 LEDs to represent the functions.

Figure 19.5. Modified X-l D Remote.

SUMMARY OF IMPACT

The X-10 Remote Controller Modification was developed for the Fircrest School in Seattle, Washington. It is intended to be used by a select group of disabled persons. The modified controller would most benefit handicapped people who have difficulty performing everyday tasks such as turning on lights or radios, and are unable to use the original 16-button remote. This in turn will allow the user to be more self-sufficient and independent.

TECHNICAL DESCRIPTION

The X-10 Remote Controller Modification is a self-contained, battery-powered device. The user interface consists of 16 LEDs and a $\frac{1}{8}''$ jack that accepts an external switch. A l-of-3 switch and two DIP switch units are also provided to allow for user configuration.

The modified controller can best be explained if it is thought of in three parts: the LED incrementer, the state machine, and the interface to the original remote control. Since the controller is to be battery powered, all CMOS components are used to minimize the power consumption. (The controller uses 0.7 mA in idle state-this will allow the 4 AA batteries to run for 3-4 months.)

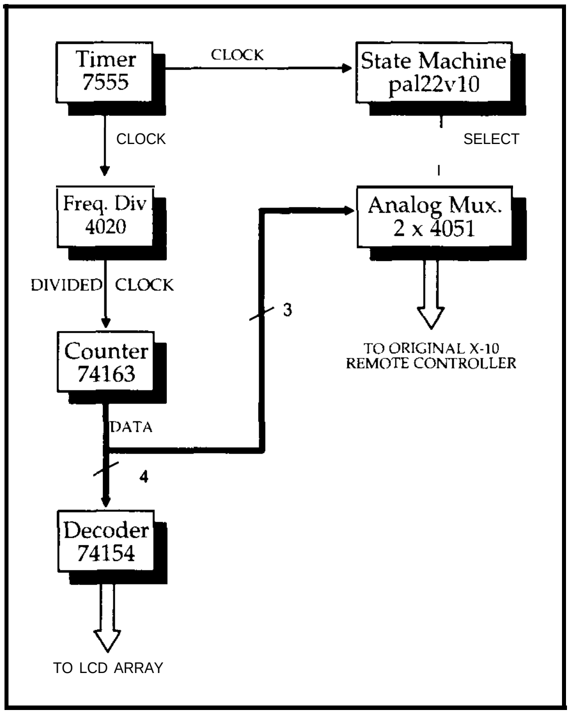


Figure 19.6. Modified X-l 0 Remote Block Diagram.

The LED incrementer uses a 74HC163 4-bit counter to cycle through numbers O-15, representing the 16 LEDs used by the controller. The output of the counter enters the select lines of a 74HC154 l-of-16 decoder. The decoder uses the signals from the counter and activates one of its 16 output lines, each of which is connected to an LED. Thus, as the 74HC163 cycles through numbers O-15, the LEDs in turn will light in sequence. Note that only one LED will be on at a given time.

The clock for the LED incrementer consists of a 7555 CMOS timer and a 4020 14-bit ripple counter. The CMOS timer is set up to output a 3 kHz clock pulse. With this clock entering the ripple counter, its frequency is divided down to one that can be used by the LED incrementer. Flexibility in the incrementer is provided by a l-of-3 switch that allows the user to select one of the three highest bits of the ripple counter. The user can choose between times of approximately 6, 3, or 1 seconds between each incrementation of the LEDs.

The 3kHz clock from the CMOS timer is also used to clock a PAL 22V10, which makes up the state machine. The state machine has five outputs, most of which properly enable and disable all other chips in the controller. There are also five inputs into the state machine, two of which are connected to a 2-DIP switch that configures the state machine. This configuration controls the number of cycles through the LEDs before turning itself off (into an idle state).

Interfacing to the original remote control relies on two 4051 8-channel analog multiplexers. Controlled by the state machine and the 74HC163 counter, the multiplexers perform the task of closing two contacts of a given button on the original board, which simulates a key press and initiates the original controller to send the proper RF signal.

The cost of reproducing the modified controller is about $56, not including the original remote. This is 1/10 the cost of a similar commercial product produced by the Prentke Romich Company, which sells a wheelchair-based X-10 remote controller interface for $525. And unlike the Prentke Romich controller, the modified remote is not limited to in-wheelchair use, as it can, for example, be used by those confined to bed.

A Self-Contained, Portable Keyboard for the Physically Impaired

Designers: Theodore Deffenbaugh, Troy Bailey
Client Coordinator: Douglas Lafever
Fircresf School, Seattle, WA
Supervising Professor: Dr. Yongmin Kim
Department of Electrical Engineering
University of Washington
Seattle, WA 98195

INTRODUCTION

The Access Macro Keyboard (AMK) is used in conjunction with the standard keyboard of any IBM XT-compatible computer. The AMK provides the user with access to 128 different macros, command sequences and frequently used words and phrases, by pressing a single key. The AMK offers a unique user interface that is simple but also powerful. The AMK allows the user to carry a small adjunct keyboard to a variety of different machines and requires no software installation for it to work.

Figure 19.7. The Physical Interface of the AMK.

SUMMARY OF IMPACT

Developmentally disabled individuals can often use ordinary microcomputers, but the rate at which they type can be a major limitation to effectively using the equipment that they own or borrow. Working with an individual that has cerebral palsy, we have developed a self-contained keyboard that will allow individuals to enter long or laborious statements and commands to any IBM PC-compatible system. Since the AMK is self-contained, requires no expansion slots, and needs no software on the host system, the AMK is ideal for those individuals that need a transportable library of keystrokes.

TECHNICAL DESCRIPTION

The AMK utilizes the Intel 16-bit 80196 microcontroller. The 80196 has high-speed output ports which allow very close wave matching to sampled waveforms. This assures that the waveform that represents a character will always be interpreted in the same way.

The 80196 is connected to an EPROM, which contains both the software necessary to run the microprocessor, and the data that is used for the macros. The microprocessor is also connected to the keyboard. Finally, the microprocessor is connected to the Intel 8279, which monitors the keyswitch array to constantly check if the 8279 indicates a key has been closed.

Although the keyboard only has 32 switches, the user can utilize 128 macros. The 32 keys have four templates that can sit over the top of the keyboard. The templates have a hole drilled in them that allows the user to rest his hand on top of the template and push a finger through the hole to contact the button underneath. The four hinged templates are specially notched so that the system can tell which template is the current one that is laying over the keyboard.

The 8279 records the status of the SHFT and CNTL pins at the time the key is pressed. These pins are used to determine the position of the overlays. Three switches along the spine of the overlay hinge are connected to produce the SHFT and CNTL signals. Switch one is closed by the first overlay, switch two by the second, and switch three by the third. The outputs of this circuit correspond to binary 0,1,2, or 3, depending on the number of overlays in place. The AMK costs approximately $150 to produce.

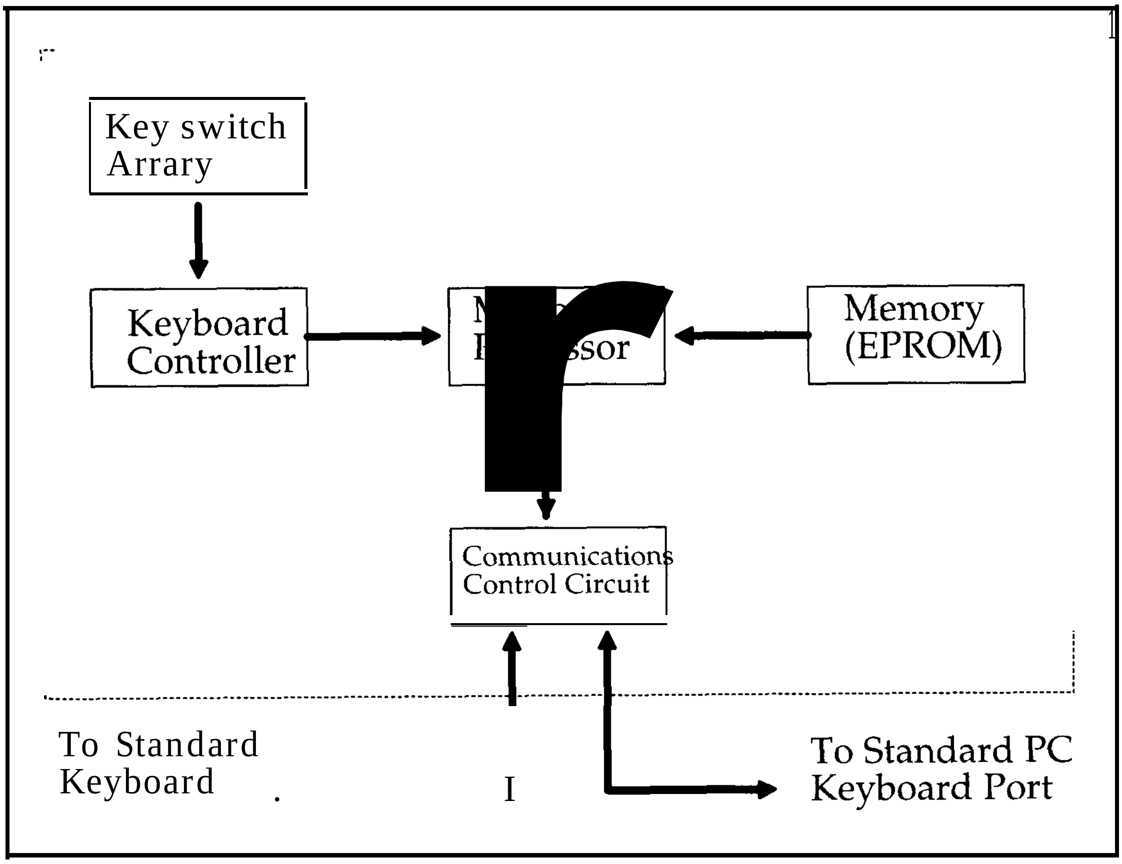


Figure 19.8. Block Diagram of the AMK.

A Device to Ease the Usability of Remote Control Devices for the Physically and Mentally Disabled

Designers: Aaron Pulkka, Vuong Do
Client Coordinator: Dr. John Eiler
Fircresf School, Seattle, WA
Supervising Professors: Dr. Yongmin Kim, Dr. Gaetano Borriello
Departments of Electrical Engineering, and Computer Science & Engineering
University of Washington
Seattle, Washington 98195

INTRODUCTION

The Programmable Infrared Transmitter with Configurable Key Pad (PITCKP) is a device designed to replace the cluttered and complex remote control transmitters utilized by physically and mentally disabled individuals with a simplified user interface configured to fit their needs.

The PITCKP is hand-held and battery powered. The size, number, and function of the keys can be set by a technician using an external programming device that encloses an infrared receiver and an alpha-numeric display. These parameters are then saved in a non-volatile storage medium for preservation in case of power loss. An overlay, indicating the boundaries and functions of the keys through words or icons, should then be placed over the flat key pad region of the device to indicate its configuration.

SUMMARY OF IMPACT

The PITCKP was designed for use by residents of the Fircrest School in Seattle, Washington, which is a residential treatment facility for the developmentally disabled. Since many of the residents have impaired motor control or cognitive abilities, the PITCKP will be used to increase the size and decrease the number of keys on their remote control transmitters for consumer electronic devices. The design of the key pad interface has been generalized to allow use with a wide variety of applications requiring key press-type input. Since the area of the key pad surface can range from that of a dime to a full table top, the possible applications range from single appliance switches to complete computer keyboards.

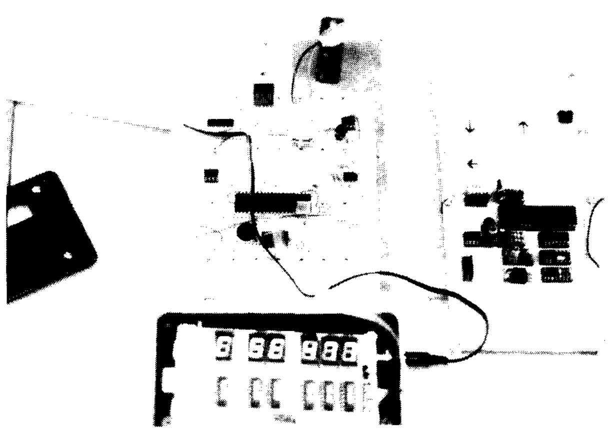


Figure 19.9. A Programmable Infrared Transmitter.

TECHNICAL DESCRIPTION

Although cost was an important consideration while designing the PITCKP, the size and power consumption limitations became the predominant constraints. The main device is housed in a 3” x 5” inch slanted enclosure (Serpac 25-S) and is powered by a single 9V battery. Wherever possible, the size, power consumption, and number of components were minimized by choosing small CMOS components. In order to maximize the lifetime of the battery, the MAX638 step-down regulator was chosen to provide a steady 5V output as the battery's output decays to as low as 6V. In addition to the power supply, the hardware design of the PITCKP can be broken into four logical modules: key pad, microcontroller, infrared I/O, and external display.

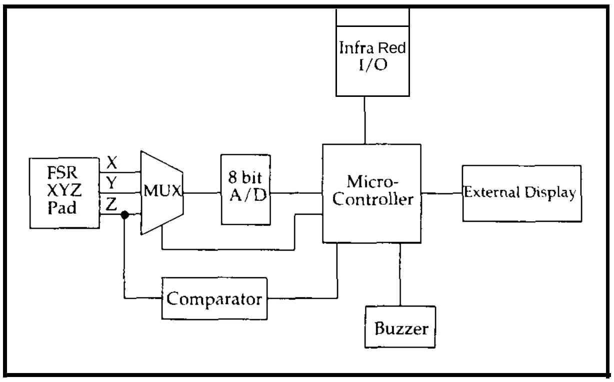


Figure 19.10 Block Diagram of PITCKP Hardware.

The key pad was implemented using force-sensitive resistor (FSR) technology. A two-dimensional FSR array (Interlink 370C) can output analog voltages representing the X, Y, and Z (force) of the applied pressure, when properly multiplexed (CD4053). Using a simple op-amp (LM358) comparator circuit, the microcontroller will receive a signal whenever a key is pressed. The multiplexed output from the key pad is then converted to an 8-bit digital value using a serial A/D converter (ADC0803).

The MC68HC805C4FN microcontroller was chosen because of its 4kbyte alterable internal EEPROM. This is used to store the device's software as well as the configuration parameters, so that an external storage device, which would have required more space and power, is not required.

The infrared output circuit was constructed using an n-channel MOSFET with noise reduction capacitors and two infrared diodes placed in series. No limiting resistor was used in order to maximize power output during the short bursts of transmission. The transmission codes are specified to meet the universal RC5 infrared transmission standard for consumer electronic infrared devices. The 36 kHz modulation required by this transmission standard was implemented in software.

The external programming device houses the infrared detector as well as the display digits used for feedback during configuration and testing. These components are only needed during the configuration process, so there was no need to house them in the transmission device itself. The display was implemented using six common anode 7-segment displays (LN514YA) driven by six SI - SO/PO shift registers (SN74HC595N). Since this device does not need to be battery powered or portable, it has its own 5V voltage regulator (7805) to step down the 9V input from an external transformer.

The software was separated into four modules that mirror the hardware structure: main, key pad, infrared, and display. The main module is separated into two modes that are entered based on the existence of the external display device. If the external display device is not present, the device enters the user mode. Otherwise, the external display is initialized and the device enters the configuration mode. While in this mode, the FSR pad is configured to accept input from the technician. The technician may then toggle between the two configuration modes: layout and program. The layout mode is used to set the number of rows and columns, as well as the gap ratio. The gap ratio represents the measurement of the dead space between keys relative to the dimensions of the key regions. Once the desired layout has been set, the technician can enter the program mode to assign 12-bit RC5 transmission codes to each key code. The key codes are numbered left to right, top to bottom, beginning with key code 1 in the upper left and key code 0 being reserved as the invalid key code. The configuration parameters are then stored in the microcontroller's EEPROM. Although it was not implemented, the hardware and software hooks are in place to read incoming infrared signals (in the RC5 format) to be stored as transmission codes for specified key codes.

While in the user mode, the microcontroller is placed into the lowest power consumption (sleep) mode waiting for the user to wake it up with an interrupt from the key pad. Once the processor is awakened, it reads the X and Y positions of the key press, maps the key press to a key code, maps the key code to a transmission code, and finally transmits the code until the key is released. Once the key is released, the processor once again falls asleep.

It is estimated that the PITCKP's main transmission device costs about $80 to manufacture, while the external programming unit costs about $15.

A Device for Assisting Memory-Lost Clients in Handling Daily Routines

Designer: Victor Nguyen
Client Coordinator: Dr. John Eiler of Fircrest School
Supervising Professor: Dr. Gaetano Borriello
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195

INTRODUCTION

The Memory Card Prompter (MCI') is a device designed to assist memory-impaired clients in handling their daily schedule. It reads a client's schedule that has been stored in a memory card, and at the scheduled time, it reminds the client of the activity that he/she needs to do by producing a sound and displaying a message on a liquid crystal display (LCD). The client can also respond to the message by pushing buttons that correspond to softkey labels on the LCD. Then, the device records the response and the time of the response into the memory card.

Figure 19.11. A Memory Card Prompter.

SUMMARY OF IMPACT

The MCP was designed for the Fircrest School, which is a school for the developmentally disabled in Seattle, Washington. The purpose of the MCP is to make life easier for the clients, especially those with memory problems. People with memory problems have a difficult time remembering even the things they do almost everyday such as brushing their teeth, eating, or taking a shower. Sometimes, this problem has a very high cost to the client. They could forget to go to work, miss a scheduled bus, or repeat an activity many times. Therefore, the MCI' was designed to help these clients by keeping track of their daily schedule, then reminding them to perform their function at the appropriate time. Although this device was designed for memory-impaired clients, it can also be used by busy people who need to be reminded of their daily activities.

TECHNICAL DESCRIPTION

The memory card prompter is implemented with these basic devices: 1) Motorola MC68HC705C8, 2) Motorola real-time clock MC68HC68T1P, 3) LCD, 4) Beeper, 5) Buttons, 6) Fujitsu 64K SRAM. The memory card prompter is implemented with the Motorola microcontroller (MC68HC705C8). This microcontroller has a total of 40 pins. These pins are divided into four basic ports. Three of the four ports (A,B,C), can be configured to be an input or output port by writing either OOH, or 11H into the direction control register of that port. For our project, port A is configured to be both an input and output port. It is used as a data bus for our project. Second, port B is configured as an output port, and it is used as a control port that sends a control signal to other peripheral devices. Third, port C is also configured as a control port. Nonetheless, **some** of the pins are unused. Finally, port D is used as a serial interface to the real-time clock chip.

Interface *with the memo y card*

The memory card is a Fujitsu 64K SRAM. However, its address line can support up to 64 Mbytes, which is in total, a 26-pin address. Thus, we do not have enough pins on our microcontroller to send out the address at the same time. Therefore, we use three latches (74HC573) to latch the address. To address the memory card, the microcontroller will break up the address into three parts and then send them out successively in three 8-bit format. After each is sent, the microcontroller uses three control pins in port B to latch that part of the address. After all parts of the address are latched, the microcontroller will set or unset the read/write line depending on whether we want to read information or write information. Then it toggles the chip-select line in port C to read or write the information. The memory card then sends information in its 8-bit data line onto the data bus. The microcontroller then latches this information into the accumulator register.

Interface with the LCD

The LCD is a device that has two lines with 16 characters for each line. The data displayed is sent out through the microcontroller over the data bus in ASCII format. The LCD has three control lines to control the interface with the microprocessor. The microcontroller sends out the data, then it sets the R/W line and toggles the enable line to register that information into the memory of the LCD. Moreover, the LCD also has a RS line that is used for the microcontroller to write the information to either its control register or its data register. The information sent to the control register tells the LCD how to display the message, and the information sent to the data register tells the LCD what to display on the screen.

Interface with the real-time clock

The real-time clock and the microcontroller communicate through the serial peripheral interface on the microcontroller. The data is sent out to the real-time clock by writing to the Serial Port Data Register then toggling the SS (Slave Select) line. When this line is toggled, the information from the microcontroller is sent to the real-time clock through the MOSI of the microcontroller and received by the real-time clock at the same line. At the same time, the real-time clock sends in the data on its data register through the MISO line to the microcontroller. The two devices are synchronized by the SCK line. The real-time clock reads in the clock from the microcontroller and uses that clock to send out its data.

The real-time clock is operated by a 32.768 kHZ crystal, and can be set to generate an alarm through its interrupt line by comparing the alarm time (hour, minute, second) with its actual time. When they match, the clock pulls the interrupt line low, and this line generates an interrupt in the microcontroller and wakes it up. The microcontroller goes to its interrupt server and services the appropriate device.

Interface with the buttons

The buttons are designed by using two 74HC74s, and a 74HC573 transparent D latch. The buttons are normally open switches and pulled high. When a button is pressed, the buttons are pulled low. This action sets one of the latches on the 74HC74 and this then automatically latches into the 74HC573. The buttons are cleared by software. The microcontroller generates a low on its clear button line, which resets the 74HC74. When the microcontroller reads the buttons, it changes the data bus to input. Then it selects the button's latch by toggling its button output enable latch. The information from the button is stored in the memory of the microcontroller.

The cost of the MCP is approximately $110.

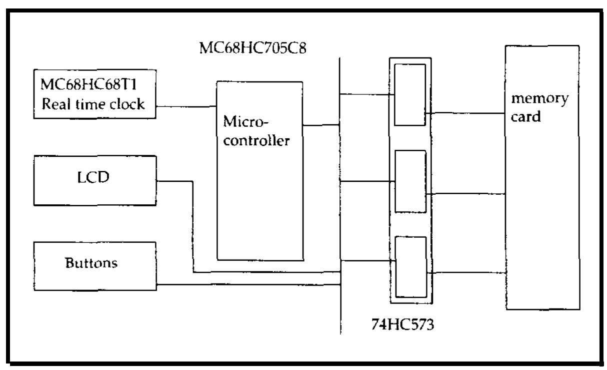


Figure 19.12. Block Diagram of Memory Card Prompter.

A System for Monitoring Work Rate and Providing Reinforcement Scheduling

Designers: Steve Stemple, Ken Cooke, Rob Spiger, Kathy Dugan
Client Coordinators: Dr. John Eiler, Robert Nansel
Fircrest School, Seattle, WA
Supervising Professor: Dr. Yongmin Kim
Department of Electrical Engineering
University of Washington
Seattle, WA 98195

INTRODUCTION

The Vocational Production Monitor (VPM) project seeks to achieve two major goals. The first is to provide a system that can aid therapists working with mentally and physically handicapped students, to help the students become independent and productive workers. The second goal is to provide an efficient means for monitoring the work done and the trend in work performance.

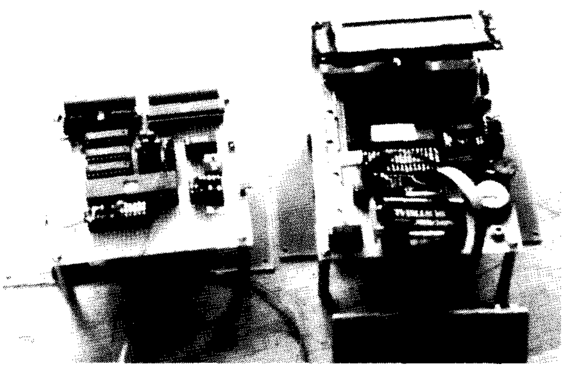


Figure 19.13. Vocational Production Monitor.

SUMMARY OF IMPACT

The VPM project design aids the therapist by automatically providing reinforcement for work performed based on a predefined, individualized reinforcement schedule. The reinforcement schedule can be defined in two ways: 1) rate-dependent, or 2) quantity-dependent. A rate-dependent reinforcement schedule will provide the reinforcer while the rate of work is greater than or equal to the minimum rate defined in the schedule. A quantity-dependent reinforcement schedule will provide the reinforcer after the completion of a specified quantity of work. In this mode, duration of the reinforcer is defined by the therapist.

The VPM is designed to monitor student work and trends in performance by storing time stamp information on an SRAM memory card. External transducers (specific to the work done) provide an input signal to the VPM each time a unit of work is completed. The time of completion and the amount of work done is stored on the SRAM memory card. The system can support memory cards with storage space up to 16 Mbytes that could easily provide storage for several months' worth of work information.

The VPM is versatile in that it can monitor students working at significantly different rates and producing different types of work. The system design ensures that a memory card is first individually configured for the student being tested. By creating a configuration file, all of the information about what type of work will be done by the student, and the reinforcement schedule that is to be followed for the job, is defined and can be stored on the memory card. Several different work activities and reinforcement schedules can be defined in the configuration file. For greatest ease of use and compatibility with available computer systems, the configuration file can be created in any text editor and is transferred to the memory card by use of the PC Interface Module, through a standard RS232C serial interface.

A memory card that has been configured with an appropriate file is ready for use in the Work Monitor Module. At the work site, the memory card is inserted into the WMM and the therapist can choose which of the jobs is to be monitored.

The VPM is designed to have the greatest flexibility in interfacing with external transducers by providing a TTL level-sensitive input for the work completed signal. The reinforcement output is designed similarly. It provides a TTL level output to control a MOSFET switch controlling the reinforcer.

TECHNICAL DESCRIPTION

The Vocational Production Monitor system consists of two modules: the PC Interface Module and the Work Monitor Module. The PC Interface Module contains the core hardware of the system, while the Work Monitor Module is a superset of that hardware containing many extra features to maintain the lowest possible power consumption.

PC Interface Module

Providing an interface between the 68 pin PCMCIA 2.0 SRAM card standard and the Motorola 68HC705 microcontroller is the primary function of this module. This is accomplished by using Port A of the microcontroller as an 8-bit data bus. Three 74HC573 latches provide a 24-bit address to the memory card. Each latch is controlled by a single I/O bit of Port B. The control signals to the memory card are provided by two I/O bits on Port C.

This module is powered from a standard 120V 60 Hz wall outlet, by a two-step power conversion. First, a 120 VAC to 9 VDC conversion is performed, and the output of this step is then regulated to 5 VDC on the module. To conserve power, this module implements a smart switch. Whenever the module is not being used (the memory card is out) the module is turned off by control logic using the Card Detect status bit from the memory card. When the card is inserted, the control logic enables the input to the 5 VDC regulator and power is provided to the board.

Transferring data from a personal computer to the memory card and from the memory card to the personal computer is the second function of this module. This is accomplished by using a MAXIM232 RS232 driver to provide the TXD, RXD and GND lines to the serial port of a personal computer. Software handshaking is provided by supporting XON and XOFF commands during serial transfer.

Software running on this module provides the therapist with several options pertaining to which data will be transferred between memory card and PC. By maintaining data pointers on the memory card, the option to transfer all data stored on the card or just the most recent data (since the last transfer) can be selected. Options for transferring the configuration file and erasing data stored on the memory card are also provided.

Work Monitor Module

This module uses the same memory card interface as the PC Interface Module. It also implements the same smart switch to provide power to the board, except that the 9 VDC input is provided by a battery. The additional hardware is discussed below.

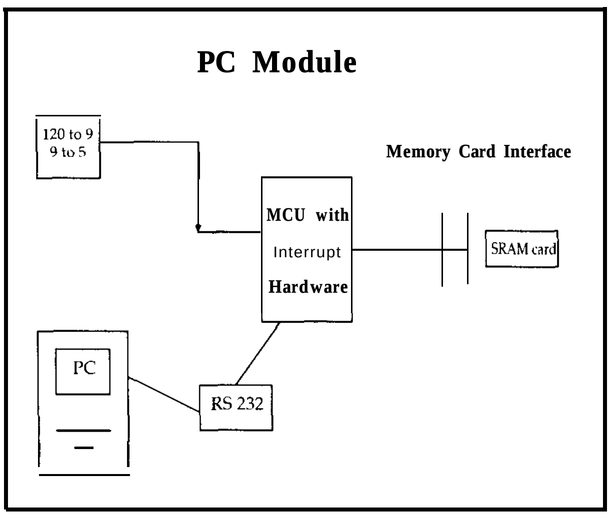
Since this module is meant to be portable, it is powered by a 9V battery. To conserve power as much as possible, a great deal of software and hardware design was completed to reduce power consumption by the module. The software is designed as an interrupt-driven processing routine. The goal is to keep the microcontroller in the lowest power 'STOP' mode as much as possible. To restart the microcontroller, an interrupt (low level on the /IRQ line) is required. This is complicated by the fact that the microcontroller only has one interrupt line and there are several sources of interrupts which are required to restart the processor.

To handle the multiple interrupt sources, combinational logic comprised of 2 CMOS NAND gates, one CMOS XOR gate and two 75HC573 latches are used to provide a single interrupt from one of the multiple sources. The NAND gates combine the multiple interrupt sources into a single output, the XOR gate and one latch provide the software a means for masking out interrupts that have already been processed. The final latch contains the single interrupt sources, so they can be read by the software to determine which source caused the interrupt, and to implement interrupt priority if more than one source provides an interrupt.

The work input to the WMM is the one interrupt source we are interested in recording as data. This is done by use of a Motorola MC68HC68T1P real-time clock. The real-time clock is used to provide the time information stored on the SRAM card. The real-time clock is battery-backed with a 3 V lithium battery. To conserve the life of this battery, a great deal of experimentation was performed to find a stable oscillation circuit for the lowest power crystal (32.768 kHz) for the clock. Data is transferred between the clock and the microcontroller through the synchronous serial port on the microcontroller.

To provide the reinforcer control, a power MOSFET is used to switch power on or off to the reinforcer. This is controlled by the software setting an I/O pin on the microcontroller. The schedule for the reinforcer output is read from the configuration file stored on the SRAM card.

User interface output is provided on the WMM by a 20x2 character LCD screen using the Hitachi LCD-II HD44780 controller. The data lines for the LCD controller are connected to the 8-bit bus on the board. Control lines for the LCD are controlled directly by I/O lines on Port B of the microcontroller. User input other than work, is done by four buttons connected to the interrupt hardware described above. Since the design for both modules is accomplished using 'off the shelf' parts, excluding the memory card, the cost of hardware is minimized. The cost of the Work Monitor Module is approximately $70. The cost of the PC Interface Module is approximately $30. An estimate of battery life for typical use (&hour day) of the Work Monitor Module is 25 days.



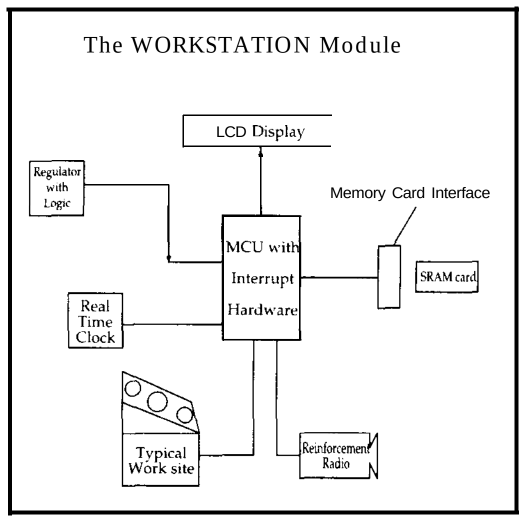


Figure 19.14. Block Diagrams of the VPM a] PC Module b) Workstation Module.