Pager Controlled Universal VCR Programmer (PCUVP)

Senior Design Project

By Jeff Shudark

Fall 1996

Advising Professor:

Dr. Keith Doty

INTRODUCTION

The goal of this project was to devise a way to program a VCR to record a show remotely, (ie: without being in the vicinity of the VCR). The programming would consist of powering on the VCR, setting the appropriate station, and initiating recording. At the proper time the recording must also be stopped and the VCR powered back down.

The functionality of actually controlling the VCR is currently done in several devices. One of these devices is the *Gemstar CT-1* Universal Remote Control with VCR Plus Technology. VCR Plus is a technology currently on the market that assigns a code to each time slot of television. These codes are readily available in a TV Guide.

This project integrates the wireless communications functionality of a pager to accomplish the goal of the remote programming. This will allow a user to program their VCR from any telephone, simply by sending a page with the program data.

CONCEPT

The equipment used in this project were as follows:

- Motorola Tango^{TM} Two-Way Pager
- Motorola M68HC11EVBU Microcontroller
- Gemstar CT-1 Universal Remote with VCR Plus Technology
- Optrex DMC40218 LCD Display (40charX2lines)

In order to accomplish the goal of using a pager to deliver a VCR Plus code, a pager with communication ability must be used. The page information must be retrieved, processed, and communicated to the VCR Plus remote.

Tango Pager Specifications:
The Tango pager was chosen because it has an integrated 3-wire RS-232 port. There is a command set defined for communication with the pager. The following is the subset of the entire command set that will be used in this project:

Command Name	Command Syntax
DownloadDeleteMessage	^y1b^y
GetStatus	^y18^y
GetOption	^y1a [param1]^y
SetOption	^y1a [param1] [param2]^y

Note: The entire command set is available from Motorol on their WWW site, It is called the Communications Linking Protocol (CLP) for personal messaging units.

These characters must be sent with at least a 40ms delay between each character. This is because the input buffers are not active until the pager is in XMODEM mode. The pager uses the XMODEM protocol to transmit and receive larger amounts of data. The XMODEM protocol will be described in the next section. During XMODEM transactions the baud rate can be sustained and the inter-character delay is not needed. However, the communication baud rate used is 1200 as to increase the integrity of the transmission.

The following is a detailed description of the individual commands used:

DownloadDeleteMessage

Description: download the next non-downloaded message from the Paging Messaging Unit (PMU).

Syntax: ^y1b^y

Protocol

Computer	Direction	PMU
^y1b^y	→	
	←	<nak> if no non-downloaded messages, or <eot> in successful
<nak>	→	(Invoke XMODEM)

Return Parameters

Parameter	Type	Description
Source	1 byte hex	Predefined address for this message
Sub address	1 byte hex	The subaddress of an Information Service message
Signiture	1 byte hex	Message identifier used for acknowledgment
Vector Information	1 byte hex	B7-4: Indicates messagecontent type 0001: Tone only 0010: Numeric 0100: Alphanumeric 1000: Hex/Binary B3: message has imbedded Multiple Choice Responses B2: RESERVED B1: message type in Information, not a personal message B0: user response for this message is requested
Time/Date	17 bytes ASCII	Time/Date message was received $HH:MM$$$MM/DD/YY where $ indicates a blank space
Data	variable size	padded with HEX 00 in necessary to 128 bytes

GetStatus
Description: This command requests current PMU status information.
Syntax: ^y18^y

Protocol

Computer		PMU	Type	Description
^y18^y	→			
	←	CannedMsgFlag	1 byte hex	0x00 - 0x0f
		TransmitMsgFlag	1 byte hex	0x00 - 0x0f
		DeviceBusy	1 byte ASCII	'0' = False '1' = True
		Out of range Status	1 byte ASCII	'0' = not out of range '1'= out of range
		Transmitter Status	1 byte ASCII	'0'=off, '1'=on
		Battery Status	1 byte hex	0 = empty up to 9 = full
		Zone ID	1 byte hex	Defined by service provider
		SubZoneID	1 byte hex	Defined by service provider
		Number of bytes available	4 bytes hex	
		Total number of messages	2 bytes hex	
		Number of non-downloaded messages	2 bytes hex	
		pad	4 bytes hex	
		PMU current time and date	10 bytes ASCII	HHMMDDMMYY
		…		

Note: The remainder of the fields are not used in this project and were therefore left out for brevity.

An inconsistency was found while implementing this command to get the current time and date. When there was a page in the pager the time and date was located at the byte offset as stated, in the 0x14 position. When no pages were in the PMU this information started at byte offset 0x12. The Motorola engineers were not aware of this problem and have been informed.

GetOption

Description: returns the device setting on the specified option

Syntax: ^y1a [param1]^y

Input Parameter:

Parameter	Type	Description
Option number	1 byte hex	index of option value to retrieve

Protocol

Computer	Direction	PMU
^y1a [param1]^y	→	
<nak>	←	<nak> if invalid option number, or <eot> if a valid option was supplied

Options

Number	Option	Values
0x00	SignitureValid *Number of days to acknowledge a message*	0x00 - 0xFF
0x01	AlertEnabled *Identifies if alerts are enabled and what the status is for AutoReadAcknowledge*.	0x00 - 0xFF Bit7-4: Reserved Bit3: AllAlerts: 0 = Disable all alerts, 1 = enable all alerts Bit2: AudibleAlert: 0 = Disable audible alerts, 1 = enable Bit1 - 0: AutoReadAck: 00 = none, 0x01 = automatic, 0x02 = manual
0x02	BaudRate Serial data communication speed.	0x01 9600 baud 0x02 4800 baud 0x03 2400 baud 0x04 1200 baud 0x05 - 0xFF 9600 baud
…		

Note: The remainder of the fields are not used in this project and were therefore left out for brevity.

Set Option

Description: updates the device setting for the specified option. Each successful call results in a PMU reset for the new option value to take effect.

Syntax: ^y1a [param1] [param2]^y

Input Parameter

Parameter	Type	Description
option number	1 byte hex	index of option value to redefine
option value	1 byte hex	new value for option number

Protocol

Computer	Direction	PMU
^y1a [param1] [param2]^y	→	
	←	<nak> if invalid option number or option value or, <eot> if a valid option number or valid option value was supplied

Options

Number	Option	Values
0x00	READ ONLY	
0x01	READ ONLY	
0x02	BaudRate Serial Data Communication Speed	0x01 9600 baud 0x02 4800 baud 0x03 2400 baud 0x04 1200 baud 0x05 - 0xFF 9600 baud
…		

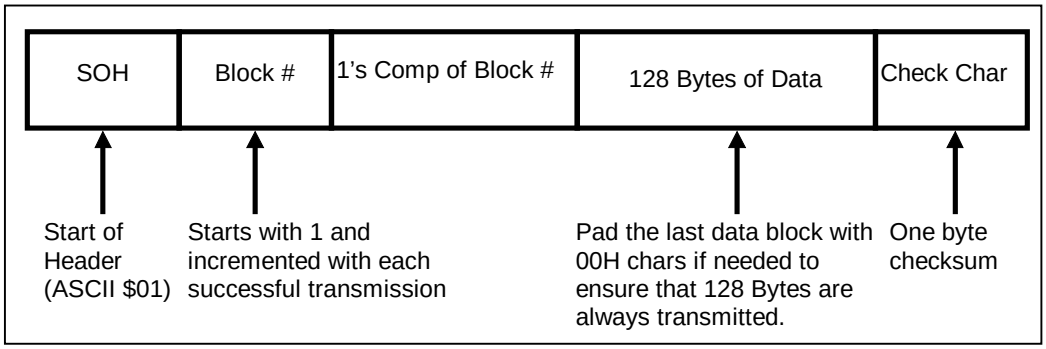
Note: The remainder of the fields are not used in this project and were therefore left out for brevity.
Note: As some options are combinations of values, the options current value must first be retrieved. The new setting must be defined and mapped onto the original value. This requires a call to GetOption prior to calling **SetOption**.
Note: A <nak> will be returned if an attempt is made to update a read only value.

XMODEM Protocol

The following is an example of an XMODEM transfer showing the details of how it is handled.

Sender		Receiver
		times out after 10 seconds
(Invoke XMODEM)	←	<nak>
<soh> 01 FE -data- <xx>	→	
	←	<ack>
<soh> 02 FD -data- <xx>	→	(data gets line hit)
	←	<nak>
<soh> 02 FD -data- <xx>	→	
	←	<ack>
<soh> 03 FC -data- <xx>	→	
(ack gets garbage)	←	<ack>
<soh> 03 FC -data- <xx>	→	
	←	<ack>
<eot>	→	
	←	<ack

The graph below shows structures of the 128 byte data block.



The input buffers of the Tango pager are operational during XMODEM transmission, and hence 9600 baud can be sustained. However, to increase integrity, 1200 baud was used for communications with the pager.

M68HC11EVBU Microcontroller

The microcontroller used in this project required additional storage. A 32 Kbyte SRAM upgrade was installed on the Microcontroller. An MC68HC24 Port Replacement Unit was also installed on the board to replace the functionality lost by installing the SRAM.

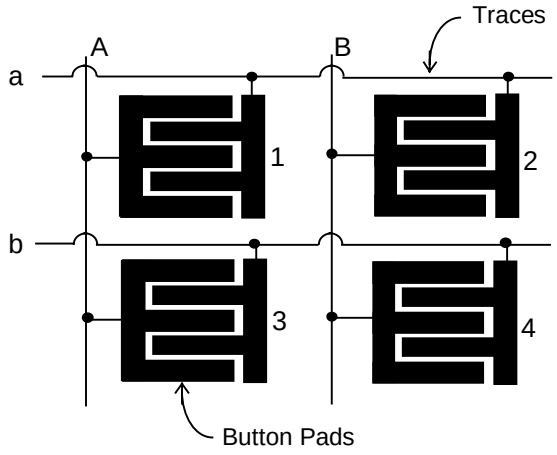
An M68HC11EVB board was originally planned for this project. This board has 16 Kbyte SRAM and the Port Replacement Unit pre-installed. When testing the communications with the pager it was discovered that the RS-232 receiver IC was incompatible with the output voltages of the Tango pager. The EVBU board was able to receive the data sent from the pager, and hence the EVBU was used.

The M68HC11EVBU also has an MC68HC68 Real Time Clock circuit installed on it. This Real Time Clock is set by issuing a GetStatus command to the pager and extraction the current time and date from the response. In the full development of the PCUVP the Real Time Clock will be used to check for due program requests.

Gemstar CT-1 Universal Remote

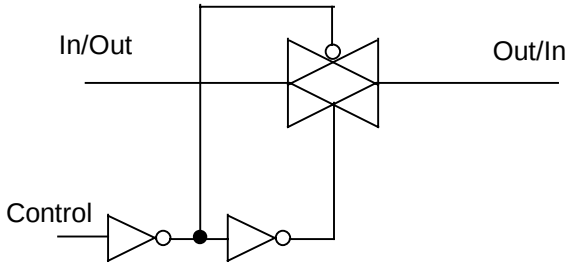
The Gemstar remote used in this project is a Universal Remote that contains VCR Plus Technology. This remote is pre-programmed with command sets for most currently available VCRs. The remote also contains an upgradeable command set. There is a circuit that allows the user to make a phone call to the manufacturer and then their system will send upgrade data over the phone line that is recognized by the remote. There is a microphone in the base of the remote for such a transaction.

For use in the project it is necessary to operate the remote from the microcontroller. To accomplish this the remote was disassembled and connections were made to the keypad to each of the control traces. The control traces are described in the following figure:



Note: When a button is pressed a carbon pad is pressed against the Button Pad making an electrical connection between a given row and column trace. As an example, when row-a and col-B are connected the 2-button is actuated.

Connections were made to all of the Row and Column traces on the remotes keypad. To make the connections to actuate the buttons 8 CD4066 ICs were used. The 4066 is a quad Analog Switch. Each switch has an input, output, and control. When the control pin is logic 1 the input and output pins are electrically connected. The circuit diagram of the switch is shown below:



Circuit Diagram of a 4066 Analog Switch

Each combination of Row and Column trace that actuated a useful button was implemented on an individual switch. Thirty-two switches were used. The control lines of each switch was connected to an output of a 4 to 16 decoder. A decoder was used because the outputs when not selected are logic 0 and not high impedance. Two decoders were used such that all 32 switches could be actuated. The following is a table showing the control line states that actuate each of the 32 buttons:

Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	HEX	Command
1	1	X	X	X	X	3X	NONE
1	0	0	0	0	0	20	0
1	0	0	0	0	1	21	1
1	0	0	0	1	0	22	2
1	0	0	0	1	1	23	3
1	0	0	1	0	0	24	4
1	0	0	1	0	1	25	5
1	0	0	1	1	0	26	6
1	0	0	1	1	1	27	7
1	0	1	0	0	0	28	8
1	0	1	0	0	1	29	9
1	0	1	0	1	0	2A	Cancel
1	0	1	0	1	1	2B	Chan ↑
1	0	1	1	0	0	2C	Chan ↓
1	0	1	1	0	1	2D	Vol. ↑
1	0	1	1	1	0	2E	Vol. ↓
1	0	1	1	1	1	2F	Rewind
0	1	0	0	0	0	10	VCR+
0	1	0	0	0	1	11	Add Time
0	1	0	0	1	0	12	Power
0	1	0	0	1	1	13	Once
0	1	0	1	0	0	14	Weekly
0	1	0	1	0	1	15	Daily
0	1	0	1	1	0	16	VCR
0	1	0	1	1	1	17	TV
0	1	1	0	0	0	18	Cable
0	1	1	0	0	1	19	Set
0	1	1	0	1	0	1A	TV/VCR
0	1	1	0	1	1	1B	Enter
0	1	1	1	0	0	1C	Record
0	1	1	1	0	1	1D	Stop
0	1	1	1	1	0	1E	Fast FWD
0	1	1	1	1	1	1F	Play

The integration board containing the decoders and switches was connected to portc of the microcontroller.

As can be seen in the table, the normal state of the port should be 3X. When a button press is desired, the associated value should be placed on the port for approximately ½ second. And then returned to 3X.

Implementation

The original plan was to do the entire programming in the microcontroller. This would require that the program requests be stored and checked at periodic intervals. If a programming request would come due before the next interval then the next interrupt would occur at the time required for that program request. At that time the program would output the commands to power on the VCR, set the proper channel, and initiate recording. A new event would then be stored to stop the VCR and power it down.

Due to time constraints the actual implementation was changed to take advantage of the abilities of the Gemstar CT-1 remote. Since the remote has VCR Plus technology that already acts as a VCR programmer, then the VCR Plus code is paged to the PCUVP.

Formatting the page:
VCR Plus codes vary in length from 4 to 8 digits. And for proper functionality the user must be able to choose from the repeating options. The repeating options are Once, Weekly, or Daily. Once means not to repeat this program request. Weekly means to repeat this requested channel/time slot every week. Daily means to repeat the requested channel/time slot every day.

To allow for this the VCR Plus page is formatted as follows:
1. The number of digits in the VCR Plus code.
2. The actual code
3. 0 or nothing = Once
   1 = Weekly
   2 = Daily

The pager is polled for new pages approximately every minute and a half. If there is a new page it is downloaded and deleted from the pager. The program information is then extracted from the page and output to the remote as a program request. The VCR Plus code is programmed by actuating the following sequence of buttons:
1. VCR Plus
2. The VCR Plus Code sent
3. Once, Weekly, or Daily

After this has been done the program is successfully in the remote and the microcontroller can resume its polling mode.

CONCLUSIONS

The Pager Controlled Universal VCR Programmer is an easy to use means of programming a VCR. It can also be used from any telephone with knowledge of the VCR Plus codes. One of the biggest advantages of this device is that if a user purchased a new VCR they would not have to learn an entirely new way to program their VCR. They would simply have to re-set the PCUVP to their new VCR and their programming will be the same as it always was. The use of the Tango Two-Way pager would also allow the PCUVP to send the programmer a response to the programming request. This could give the programmer feedback as to weather the programming request was:

- valid
- invalid
- successful
- unsuccessful
- conflicting with an existing program
- etc

The PCUVP is an easy to use means of attaining the ability to program your VCR to record a show from any telephone.

Prototype of the PCUVP: