

EVALUATION OF THE SUPPORT EFFICIENCY OF HARDWARE CONTROLLERS FOR MULTI-DIGIT 7-SEGMENT INDICATOR DISPLAYS

Borislav Petrov Radoslav Raychev

Abstract: Hardware controllers for multi-digit displays with 7-segment indicators are described at architectural and interface levels. A generalized program presents the basic interactions between the host and hardware controller in the terms related to peripheral accelerators. The obtained theoretical and practical estimations of the CPU time spent to manage the peripheral accelerator for 7-segment indicator displays illustrate their support efficiency.

Key words: CPU time utilization efficiency, peripheral accelerator, 7-segment indicator, microcontroller.

INTRODUCTION

Seven segment indicator (7SI) displays are still popular in human-machine interfaces (HMI) [1, 2, 11]. Connecting the host microcontroller to an R-digit 7SI display is a problem with broad consequences at theoretical and application level (fig. 1).

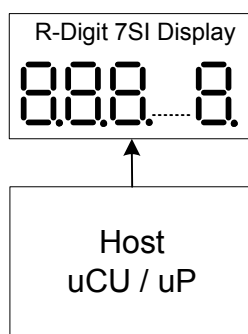


Fig. 1. Connecting the host to a R-digit 7SI display

The first solution consists to build individual links to each of R 7SI digits in star topology. In this case, it is difficult to deal with R exceeding 2 to 4 display digits because the number N of connecting lines grows in linear function of R: $N = 9 \cdot R$ (8 lines to segments, including a line to the decimal point (DP) segment and a common anode/cathode line).

The "9*R-line problem" is moderated by attaching all R indicator digits to a common 8-line "select-N-segment" bus, $0 \leq N \leq 8$, or NS-bus and to R separate "select-a-digit" lines. This NS-bus solution implements the time-multiplexing method between R digits (R-TMM). For large displays with a number of digits $R > 8$ it is possible to implement time-multiplexing method between 8 segments (S-TMM). In those cases, it is preferable to put into operation a "select-M-digits" bus or MD-bus, $0 \leq M \leq R$, and 8 separate "select-a-segment" lines. Both R-TMM and S-TMM are sometimes called "dynamic indication". A time-multiplexing is more commonly implemented in software and not in hardware because of it cost saving. But it exploits significant part of CPU time. Moreover, it is difficult to increase the digit numbers R for the R-TMM without compromising the visualization quality, as the display's refreshing frequency lows down when rising R. Alternatively, the S-TMM displays drawback is the consumption. It could go up to $R \cdot I_{\text{seg}}$, (I_{seg} is the segment current), in contrast of the consumption up to $8 \cdot I_{\text{seg}}$ for the R-TMM.

For this reason, special integrated circuits, implementing the dynamic indication for 7SI in hardware, have been developed as peripheral accelerators (PAs) for microcontrollers (uCU) and microprocessors (uP) [3, 5 - 11]. The PAs, substantially modernized, are now incorporated in the uCU chips of many chip makers (fig. 2). They maintain the autonomy of operation after a configuration made once by software and free the CPU for other tasks. They work in parallel with the CPU, keeping the refreshing frequency and supply current constant and independent of the software transactions, thus shifting the burden from software to hardware.

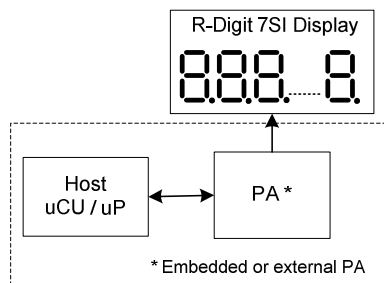


Fig. 2. Connecting the host to a display through PA

This work evaluates a common solution for dynamic indication based on a generic PA. It aims to evaluate the PA's support efficiency in terms of CPU execution time ratio.

STRUCTURE OF A PERIPHERAL ACCELERATOR (PA) FOR 7SI

To understand the support efficiency of a PA for multi-digit 7-segment indicator displays it is mandatory to considerate its common and particular characteristics.

All available PAs [3, 6 - 10] have in common the R-TMM capability, hence aim the restriction of display power consumption. It is supported by integrated oscillator(s), by voltage/current level translation power drivers, by one or more character font generator(s) in ROM or RAM and by controlled via software blanking display. Additionally, all they have some particular characteristics presented in Table 1 and fig. 2.

Table 1. Particular characteristics of popular PAs

Attribute	PA	MAX7219 [3]	SAA1064 [6]	ICM7228 [7]	MM5450 [8]	M5482 [9]	MAX7211 / 7212 [10]
Number of supported 7SI digits		8	4	8	4or5	2	4
Daisy-chaining capability		Yes	No	No	No	No	-
BCD-to-7 segment decoder		Yes	No	Yes	No	No	Yes
Character font number		2	-	1	1	1	2
Interface with the Host		3-wire serial	I ² C	3-wire serial	3-wire serial	2-wire serial	8-wire parallel
Bits in a data frame		16	40	8	36	36	4+4 control
Brightness control: analog		Yes	No	No	No	Yes	Yes
digital (PWM)		Yes	No	No	No	No	Yes
Display test		Yes	Yes	No	No	No	Yes
Package type		24 pin SOIC	24 pin SOIC-DIL	28 pin SOIC-DIL	40 pin DIL	20 pin DIL	44 pin PLCC

The maximum light intensity of segments is imposed in some PAs by the Analogue Brightness Control circuit, adjusted by means of external resistances. It can be furthermore reduced by Digital Brightness Control circuit via data writes in a kind of Brightness Register. Written data determines the PWM (Pulse-Width Modulation) ratio and the brightness shrink. The number R of connected 7SI must be communicated to the PA too, as it determines the display quality also.

Data registers, containing the code of the symbol that must appear on the corresponding (one from R) 7SI are placed in an R-byte dual-port SRAM memory (fig. 2).

Three-wire interface can be easily implemented in any microcontroller.

Several PA have pins used for assuring daisy-chain connection and obtaining displays with increased number of digits.

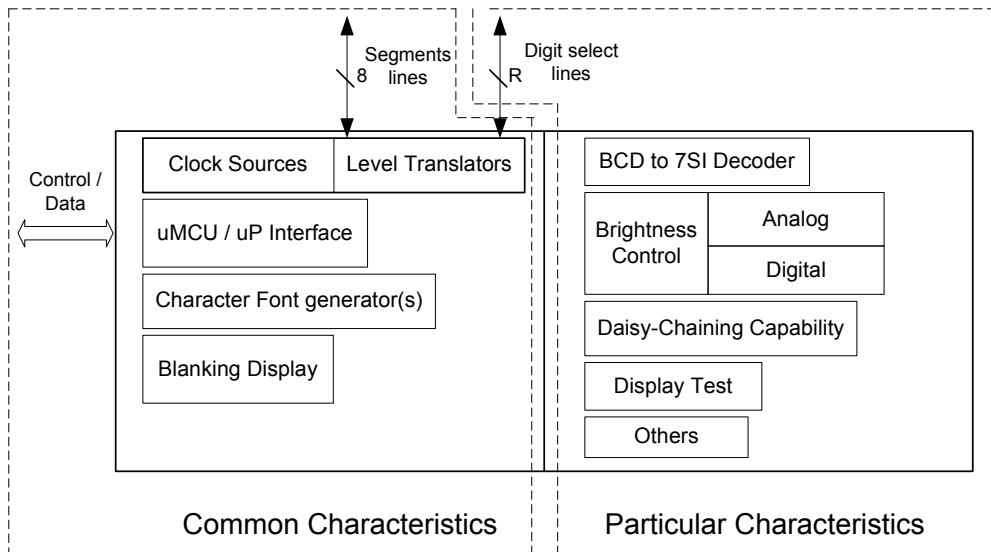


Fig. 3. Generic structure of a PA

Number of transactions "host-controller"

Fig.4 represents the general form of an algorithm depicting interactions between the host (uCU / uP) and a PA. The obligatory PA, display and application resources initializations are performed only once after power-on (Fig. 4a). It is clear (Fig. 4b), that the execution time for the "Servicing of PA modes or digit settings" operator is not performed in all iterations of the endless system loop. It is important to point out, that the program execution passes through the "NO" branch of the IF operator (the diamond shape) more frequently, than via the alternative "YES" branch. Moreover, the operator's execution time $T_{SRVC} = T_{MSG}$ of the "Yes" branch is short in general.

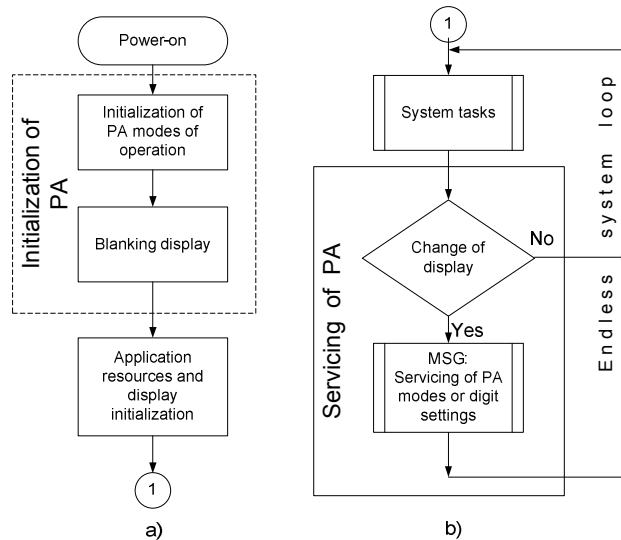


Fig. 4. Algorithm of interaction between the host and the PA

Let T_{TOTAL} is the total execution time for l iterations through the endless system loop, T_{SYSTEM} is the execution time for "System tasks" operator per iteration, and k is the number of "Yes" path exits via the IF operator in the time period for $l \geq k$ iterations of the endless system loop, so k is the "host-controller" transactions number in l iterations. Then:

$$T_{TOTAL} = l * T_{SYSTEM} + k * T_{MSG}. \tag{1}$$

In a typical HMI scenario, the display contents changes needed per second is inferior of 10, so $2 \leq k < 10$ for $T_{TOTAL} = 1$ s. For this reason, the use of the PA is favourable for the processor load, because the time $T_{MSG} \ll T_{SYSTEM}$ and even $(2 \dots 10) * T_{MSG} \ll T_{SYSTEM}$.

Implementation

To clarify the basic service time relationships in the case of R-digit 7SI display (Fig. 5), a MAX7219 [3] peripheral accelerator was prototyped in the development system STK500 [12] with an 8-bit Atmel AVR ATmega8515 microcontroller [13]. The code writing and debugging was accomplished in Atmel's AVR Studio IDE [14].

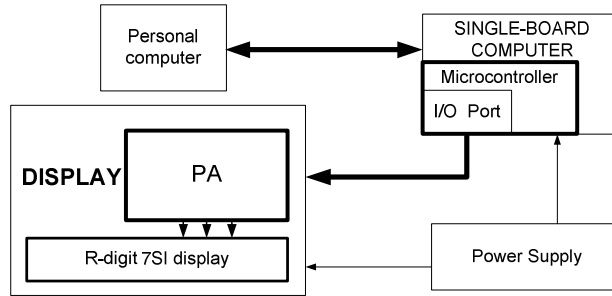


Fig. 5. Block diagram of the evaluation system

Algorithms for PA servicing

A typical implementation of the algorithm uses the synchronous serial interface SPI built in many microcontrollers. A less time-efficient version wraps the general purpose peripheral parallel interface pins (Fig. 6). It is the preferred one for worst-case estimation.

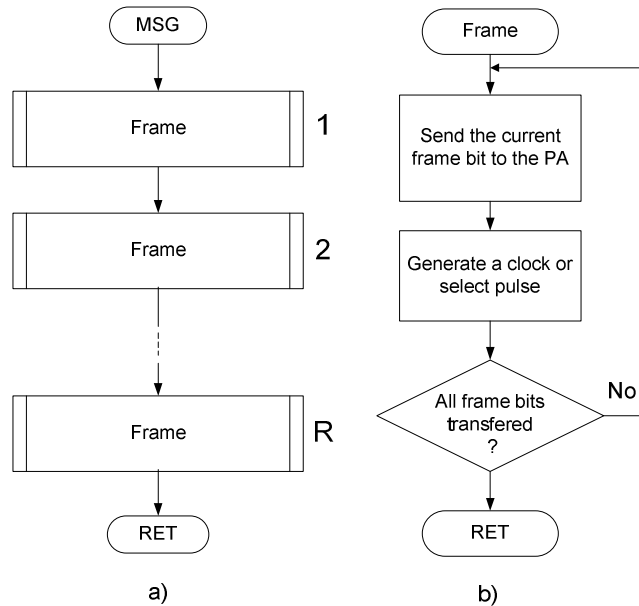


Fig. 6. Block diagram of the PA servicing algorithm

In the common case, the MSG subroutine sends to the MAX7219 PA up to R frames to refresh $n \leq R$ 7SI (Fig. 6 (a)). Each frame is serialized by the software on a bitwise basis by the Frame procedure. A clock pulse, generated in software too, accompanies each frame bit, as required by the PA manufacturer [3] (Fig. 6 (b)).

The algorithm of Fig. 6 (a) is presented in a general form and this admits interesting interpretations for the T_{MSG} execution time. Sometimes it is necessary to refresh only $n < R$ digits. Then the T_{MSG} decreases compared to the T_{MSG} maximal value. In other cases, the HMI may involve display effects changes, such as indicators start / stop blinking, the

brightness increasing / decreasing and so on. The reprogramming of the control registers for these purposes usually increments the T_{MSG} value.

DISCUSSION

The ATmega8515 uCU in the STK500 development board is clocked at a frequency of $F = 3.69$ Mhz, and the cycle period is $T_{CYC} = 1/F = 272$ ns. The physical measurement of the send-a-frame time T_F is about $111 \mu s$. Then the T_{MSG} value is

$$T_{MSG} = (n \leq R) * T_F \leq R * T_F = 8 * 111 \mu s \leq 1 \text{ ms.} \quad (2)$$

In the case of $2 \leq k < 10$ for $T_{TOTAL} = 1$ s, the $T_{MSG} \leq 10$ ms in one ms, or lower than 1 % of the CPU execution time. The support efficiency of PA for multi-digit 7-segment indicator displays is obviously positive.

CONCLUSIONS

Although the estimations presented here refer to a given uCU clocked at a given frequency, the judgments are applicable in the common occasion. The scenarios for displays with $R = 8$ or fewer 7SI per PA are easily scalable, because the implementations with daisy-chained PAs for $8 * R$ digits will load the CPU slightly more, keeping on the PA servicing time in the range of few percents for regular practical values of digit numbers.

ACKNOWLEDGMENTS

This paper is financed by project: Creative Development Support of Doctoral Students, Post-Doctoral and Young Researches in the Field of Computer Science, BG 051PO001-3.3.04/13, EUROPEAN SOCIAL FUND 2007–2013r. OPERATIONAL PROGRAMME “HUMAN RESOURCES DEVELOPMENT” and the TU-Gabrovo research project E906-2010.

REFERENCES

- [1] LEDs Are Still Popular. Application Note AN1183. Maxim Integrated Products. <http://www.maxim-ic.com/> .
- [2] Electronic Displays Comparison. Maxim Integrated Products. Application Note AN1193. <http://www.maxim-ic.com/> .
- [3] Serially Interfaced, 8-Digit, LED Display Drivers MAX7219, MAX7221. Maxim Integrated Products. www.maxim-ic.com
- [4] Data Multiplexer Adds Cursor To MAX7219 or MAX7221 LED 7 Segment Display Drivers. Maxim Integrated Products. Application Note AN141. <http://www.maxim-ic.com/> .
- [5] Using the MAX7219/7221 to Drive Higher Voltage or Current. Maxim Integrated Products. Application Note AN1196. <http://www.maxim-ic.com/> .
- [6] Four-digit LED-driver with I2C-Bus interface SAA1064. Philips Semiconductors.
- [7] 8-Digit, Microprocessor-Compatible, LED Display Decoder Driver. Intersil ICM7228 Data Sheet. <http://www.intersil.com/> .
- [8] MICREL 8-Digit LED Display Driver MM5450. Micrel Semiconductor, Inc. <http://www.micrel.com/> .
- [9] LED display driver M5482. ST Microelectronics. <http://www.stm.com/> .
- [10] Four-Digit Display Decoder / Drivers ICM7211, ICM7212. Maxim Integrated Products. <http://www.maxim-ic.com/>
- [11] Craig Hollabaugh, Embedded Linux: hardware, software, and interfacing. Addison-Wesley, 2002.
- [12] Atmel STK500 Starter Kit User's Guide. Atmel Corporation. <http://www.atmel.com/> .
- [13] Atmel 8-bit Microcontroller ATmega8515. Atmel Corporation. <http://www.atmel.com/>.

[14] Atmel AVR Studio. Atmel Corporation. <http://www.atmel.com/> .

ABOUT THE AUTHORS

PhD student Borislav Petrov, Department of Computer Systems and Technologies, Technical University of Gabrovo, Phone: +359 66 827 501, E-mail: petrov_br@tugab.bg.

Assoc.Prof. Radoslav Raychev, PhD, Department of Computer Systems and Technologies, Technical University of Gabrovo, Phone: +359 66 827 597, E-mail: raychev@tugab.bg.