AN ALTERNATIVE METHOD OF INTERFACING A DC MOTOR TO A MICROPROCESSOR FOR MOTION CONTROL.

M.F.Selekwa

Dept. of Mechanical Engineering, University of Dar es Salaam,
P.O.Box 35131, Dar es Salaam, Tanzania

ABSTRACT.

A Simple and cheap alternative method of interfacing a permanent magnet dc motor to a microprocessor(MPU) for motion control using switching transistor motor drivers is presented. This method involves the use of very few transistor-transistor logic (TTL) devices and a 555 timer to produce a pulse width modulated (PWM) signal which is required by the motor driver. The motor driver in this case should be of an H-bridge type. Like other methods currently in use, it relieves the MPU of all timing functions which are necessary in the generation of PWM signals. The present method, however, is simple and cheaper than the conventional methods.

Laboratory test results using an 8-bit Motorola 68HC11 microcontroller showed that the interface works very well. It is however limited to 8-bit microprocessors only. Further work should now be directed towards enabling it to work directly on microcomputers by increasing the number of input lines to 16 or 32 and inclusion of input-output devices such as the peripheral interface adapter and an address decoder.

INTRODUCTION.

Since alternating motors are difficult to control because of their non-linear characteristics [1], most engineering real time systems which involve motion use permanent magnet d.c. motors as their prime movers. The motion of a dc motor is controlled either by field current or by the armature voltage (equation 1). In the case of armature voltage control, the higher the armature voltage, V_a, the higher is the motor speed ω_m, and at zero V_a the speed also comes to zero.

$$\omega_m(t) = \frac{V_a(t) - R_a i_a(t)}{K_b} \qquad (1)$$

Where K_b is the back emf constant, R_a the armature resistance and i_a the armature current.

The armature voltage is supplied by motor driver amplifier. There are two common types of amplifiers employed in controlling the dc motor; the linear and the switching transistor types. The linear transistor type amplifies analog signals fed to its input and then fed to the motor. This needs a digital to analog converter (DAC) to process MPU signals being fed to a motor. The switching transistor type is driven directly by digital - signals - to switch ON and OFF the power to the motor continuously to control the mean armature voltage.

Of these two amplifier types, the switching transistor is more common because of its power capabilities [2]. One example of the switching transistor amplifier is the IR8200B, a 3 Amp 55 Volt DMOS H-Bridge power integrated circuit used in the interface being presented. The amplifier has two modes of operation; the first known as pulse frequency modulation (PFM) has fixed duty cycle and variable frequency (Fig.1 (a)) while the second type known as the PWM has a constant frequency and a variable duty cycle(Fig.1 (b)). Because of its simplicity in realizing practically, PWM is more popular in real time systems applications than the PFM.

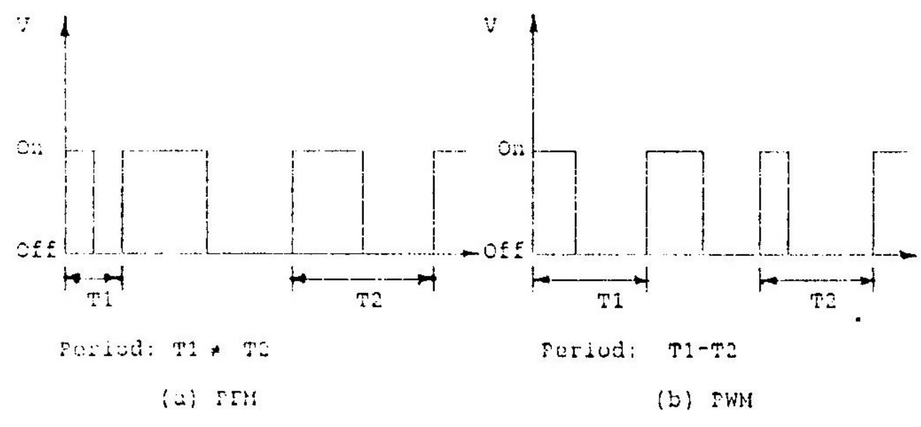

Fig.1: The two modes of operation of the switching transistor amplifier.

The method of converting the microprocessor signal to be able to switch ON and OFF the switching transistor motor driver has for a long time been a bit expensive. This is due to either MPU code generation time and usage by letting it do all the necessary timings required for PWM signal generation, or by incorporation of expensive interfacing hardware to take care of all the timing requirements.

In the first case, the MPU is programmed to keep switching ON and OFF one bit on its output port. In addition to other tasks, the same microprocessor has to take care of all the timing requirements of the PWM signal as shown in the following example for the Motorola 6802 microprocessor and the 6821 peripheral interface adapter (PIA). The calculated control effort (V) is assumed to be stored at address $1000 and the PIA base address is $2000. Line CB2 of the PIA is used for PWM signal output.

```
            .
            .
Initialize the PIA
            .
            .
PWM LDAA #3C      ;Set ON
    STAA 2003
    LDAA 1000
ON  DECA
    BNE ON        ;Delay for V MPU cycles
    LDAA #34      ;Set OFF
    STAA 2003
    LDAA #FF
    SUBA 1000
OFF DECA          ;Delay for $FF-V
    BNE OFF
    BRA PWM       ;Start another cycle
```

This approach is undesirable, particularly in multi-tasking real time systems where there are other tasks, like reading sensors and doing calculations which also need the attention of the same MPU. If the program is not to be very complicated, this approach needs two dedicated microprocessors for the motor control; one microprocessor for reading sensors and doing all control calculations and another for generation of the PWM signal accordingly. This is rather an expensive approach.

The second approach can be implemented in several different methods; only the most popular methods will be discussed here, i.e using the voltage-to-frequency converter (VFC) and the use of special motor motion control integrated circuits (ICs).

In the first method, a binary (digital) signal from the MPU is fed to the DAC to get an analog voltage corresponding to the digital output of the MPU. Since this analog is not sufficient to drive a dc motor because of the power limitations of the DAC [2], it is then amplified via the VFC which changes the dc analog signal to a frequency modulated (FM) one. Direct amplification of this analog signal using linear amplifier is also possible, however, due to power limitations of the linear transistor amplifiers as mentioned earlier, amplification has to be done using switching transistors, therefore calling for an FM signal. This FM signal is the one that is fed to a PWM circuit which is a one shot (pulse) module timer and a counter. The counter records the rising edges of the incoming FM signal in a specified period of the required PWM signal with the final count setting the width of the pulse. A train of these pulses results in the required PWM signal. The DAC, VFC, and the PWM circuit together form the interfacing hardware for motor control (Fig.2).

The second method uses special ICs such as the Hewlett-Packard HCTL1100 and the National Semiconductor LM628/LM629 motor controllers [1] as interfacing hardware (Fig.3). Generally, these ICs work almost in the same way as described above, i.e accepting a digital signal and generating a corresponding PWM signal.

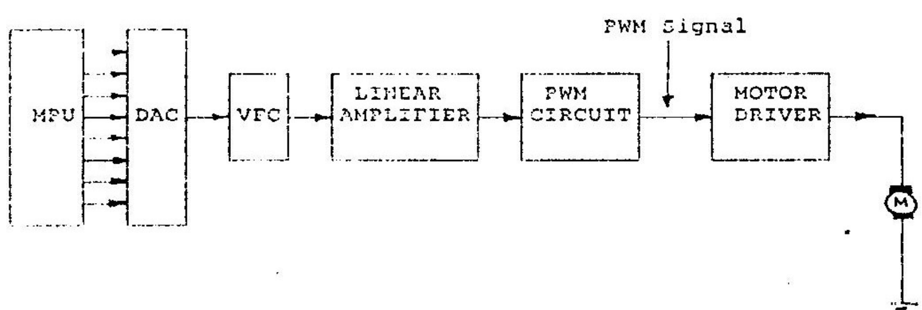

Fig 2: Combination of the DAC, VFC, and PWM circuit as an interface.

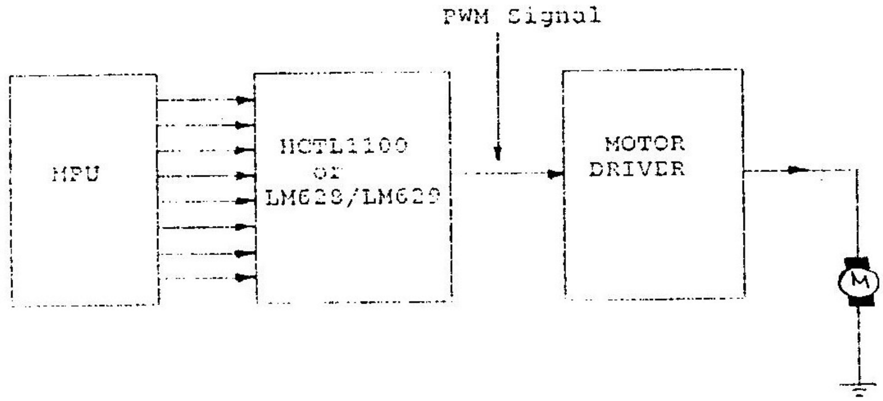

Fig.3 Interfacing using special ICs.

Both approaches discussed so far relieves the MPU of all the timing functions as it views the interfacing hardware as a byte of memory and hence it only writes the required drive voltage in that byte then continuing with other tasks. the following pseudo code example shows how the microprocessor may be programmed to use these hardware. The interface address is assumed to be MOTOR .

```
initialize;
LOOP: read sensors;
      calculate control effort;
      put the calculated control effort in
      MOTOR;
      wait for the next sampling time;
      repeat  LOOP until STOP;
  end;
```

On the other hand, these methods are rather expensive, particularly when one considers the amount of hardware to be used. For example the HTCL 1100 IC alone costs not less than U$ 300/= (1993 Australian prices), the same is for the combination of the DAC, VFC and the PWM circuit. The alternative method presented in this paper reduces these costs by using simple and cheap components, yet it offers the same advantages as these conventional methods.

THE ALTERNATIVE DC MOTOR INTERFACE.

Structure and Principle of operation.

The proposed dc motor interface also works on the principle of PWM by producing a signal whose duty cycle is proportional to the required motor armature voltage. It accepts a byte of binary signal representing the drive voltage from the MPU and converts it directly to the a corresponding PWM signal without passing through the analog stage. Its main advantage is that it uses simple logic devices for decoding the MPU digital signal to the appropriate motor voltage [3,4].

There are five main parts in the interface (Fig.4); the input circuitry, timer, PWM pulse generator, polarity (direction) decoding logic and the output circuit. To simplify its application, the output circuit has a built in motor driver which is the IR8200B, a 3 Amp 55 Volt DMOS H-Bridge power integrated circuit.

Two Quad-Optoisolators (ISQ74) and a simple network of resistors are used in the input circuitry to separate the microprocessor from the high voltage power side of the motor yet maintaining the TTL HIGH and LOW levels of the input signal. The input signal is a signed 8-bit byte from the MPU, in which the magnitude is coded in the 7 least significant bits (LSB). The most significant bit (MSB) is used for coding the polarity of the signal.

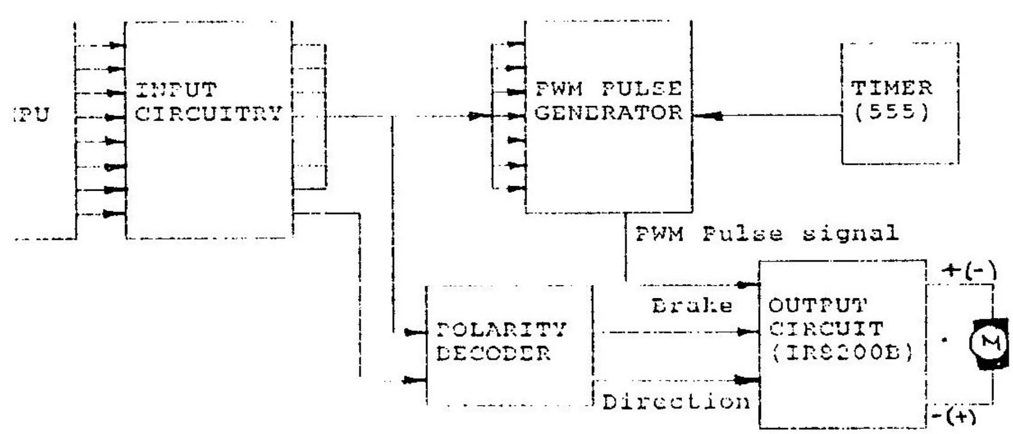

Fig.4: Functional layout of the proposed Interface.

The timer is an astable multivibrator whose function is to generate timed constant pulses which are then used to drive the time function of the PWM signal generator. The desired PWM signal frequency (PWM freq.) and the timer frequency (TF) are related by:

$$PWM.freq. = \frac{TF}{127} \quad (2)$$

Where 127 is the number of timer pulses required to overflow the 7 LSB. Since the PWM frequency needs to be above the audible range in order to ensure smooth motor drive, the timer frequency has to be more above the audible range. Because of its low cost, a 555 timer IC connected as an astable multivibrator with a frequency of 160 kHz is used. To make a clear distinction between the HIGH and LOW levels of the generated pulses, a Schmidt trigger (74LS14) is included in the multivibrator output to sharpen the pulses [5].

There are two main parts in the PWM pulse generator; a binary counter and a comparator. A dual 4-bit binary counter (74LS393) is connected to give a 7-bit count of the incoming pulses from the timer unit. Basing on the timer frequency of 160 kHz, the counter overflows at a rate of 1260 Hz. This overflow rate is the base frequency of PWM signal generated and which remains constant. A magnitude comparator (74LS684) is connected as to receive the instantaneous count of the binary counter on its P-side and the 7 LSB of the MPU digital signal on its Q-side. Its useful output is taped from pin $\overline{P > Q}$, i.e pin 1, which is always HIGH when the count on the P-side is less than that on the Q-side, otherwise it turns LOW [6]. It can be seen that, this output will be alternating between HIGH and LOW at a constant frequency of 1260 Hz corresponding to the counter overflow rate. However the duty cycle, i.e the ratio of the period when the output is HIGH to the overall cycle period will depend on the magnitude of the MPU digital signal fed to the comparator. This is what is required for a PWM signal.

Although the digital signal polarity is simply coded in the MSB of the signal, the polarity (direction) decoding network has been designed to take care of the neutral direction whereby all bits are either HIGH or LOW. Whenever the neutral direction is detected a braking signal is activated.

Two 8-input positive NAND gates (74LS30), two octal bus transceivers (74LS640) and a quadruple 2-input positive NAND gate (74LS00) are used in the network. The transceivers have their control inputs biased such that they invert input on side $\overline{A}$ to side $\overline{B}$, i.e are made inverters. Only one transceiver could have worked, but because of the pull up resistors which are to be included on each of the input lines from the opto-isolator for signal stability but whose effects have to be filtered from the input digital signal, two transceivers are used. The output of each of these inverters is NANDed using the 74LS30 TTLs whose outputs are again NANDed to determine the polarity of the braking signal. The braking signal is selected so that it is LOW for the motor to run otherwise the motor stops (i.e brakes). Since the braking signal is dual natured, i.e can result from all the digital signal bits being either HIGH or LOW, it becomes necessary to have an inverter and two 8-input NAND devices mentioned before.

The basic interface output is just three signal lines: the PWM pulse line which comes from the pulse generator, the brake line and the direction line from the direction decoding network. Since the H-Bridge motor driver (IR8200B) is incorporated in the interface output, the three basic output lines are fed to the driver whose output has only two lines, i.e. the negative and the positive lines. These two lines alternate in polarity depending on the required motor direction.

The total cost of all these components is less than $200/= (according to 1993 Australian prices) including the motor driver power IC, this offers a relatively cheap option for MPU interfacing of dc motors as compared to the conventional methods.

Interface performance testing and results.

This interface was tested in a laboratory using Motorola 68HC11 microcontroller. The microcontroller was programmed to feed the interface with different digital signals and the resulting motor armature voltage and speed were recorded using a digital voltmeter and an optical tachometer respectively. The results in table 1 were obtained using a motor power supply voltage of 20 V DC 3A on a motor rated 24 V DC 3 A, 10,000 rpm.

Table 1: Performance results

MICROPROCESSOR DIGITAL SIGNAL	DECIMAL EQUIVALENT	ARMATURE VOLTAGE [V]	SPEED [RPM]
00010100	20	3.07	1019
00100110	38	6.05	2165
00111001	57	9.04	3331
01011111	95	15.01	5675
01110011	115	18.03	6882

These results were plotted for the digital signal and the armature voltage as well as the motor speed and showed a linear relationship as shown by the plots in Fig.5. Generally the interface was found to work very well for controlling motors with power requirements of up to 3A and 55V which the IR8200B motor driver can hold.

DISCUSSION

Linear regression analysis of these results shows that there is a linear relationship between the armature voltage V_a and the decimal equivalent of the digital signal (DE) given by:

$$V_a = 0.1575DE \tag{3}$$

Errors of ±0.0768V and ±0.0005 in the armature voltage readings V_a and the DE coefficients are observed. This relationship compares to the theoretical one which is:

$$V_a = K.DE \tag{4}$$

where K is a constant of proportionality.

Similarly, the linear relationship between the motor speed (N) and the decimal equivalent of the digital signal is found to be:

$$N = 61.66DE - 193.4 \quad (5)$$

The corresponding standard errors in the speed N and the coefficients of DE are ±19.20 rpm and ±0.244 respectively. Though experimental and statistical errors are present in both equations (3) and (5) above, these equations, however, resemble to theoretical ones where equation (5) is similar to equation (1) for the motor voltage and speed relationship.

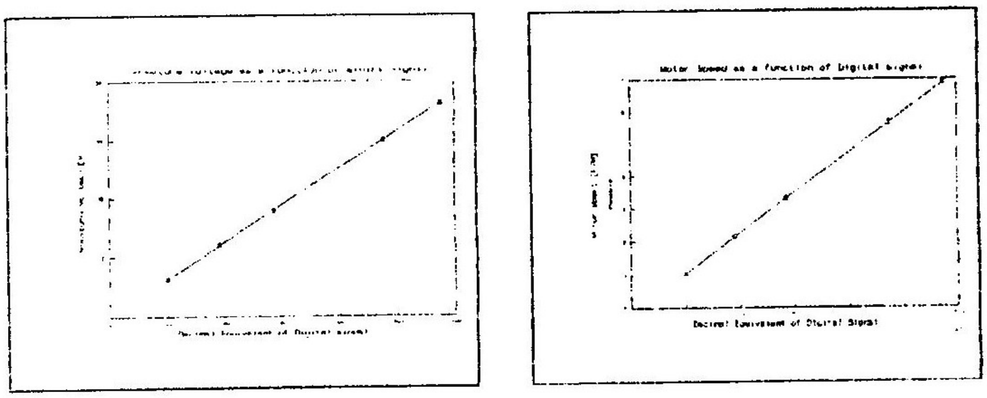

Fig.5: Plots showing performance results

CONCLUSION

An alternative cheap hardware for interfacing a dc motor to a microprocessor has been presented. The principle of its working has been described comprehensively. It uses cheap components to drive powerful dc motors with performance comparable to the commercially available chips which are relatively expensive. The whole hardware design is simple and can be constructed easily even in a simple laboratory.

It is noted, however, that unlike commercial chips, the reference frequency of the signal in this case is subject to fluctuations depending on how temperature changes affects the 5.6 kΩ resistor and the 0.001 μF capacitor across pins 6, 7 and ground of the 555 timer (Appendix). These passive components in the timer circuit are the ones that determine the signal frequency. In most practical cases, changes in timer frequencies will have no effects to the overall interface performance because the rate of change is normally slow so that the PWM signal frequency becomes almost constant.

ACKNOWLEDGEMENTS

The author wishes to express sincere gratitude to his supervisors, Dr. Michael Tordon and Mr. Jim Sanderson of the Department of Industrial Automation, School of Mechanical and Manufacturing Engineering, University of New South Wales for their invaluable contribution to the success of this project.

REFERENCES

1. Gaykward, R.; Sokolof, L.:Analog and Digital Control Systems. *Prentice Hall International*, 1988.
2. Stiffler, A.K.: Design with Microprocessors for Mechanical Engineers, *Mc-Graw Hill* International Editions, 1992.
3. Selekwa, M.F.: Design and Implementation of a Computer Control System for an unstable Mechanical System: case of an inverted pendulum. M.Eng.Sc. Thesis. University of New South Wales, Sydney, Australia. 1993
4. Steward, L.J.: Control System design for a broom balancing car. B.E Thesis. University of New South Wales, Sydney Australia 1991.
5 Tocci, R.J.: Digital System: Principles and Applications. *Prentice Hall International,* 1986.
6. Floyd, T.L.:Digital Fundamentals, 3rd Edition. *Charles E.Merril Publishing Company*, 1986.