

Controlling a Brush DC Motor with an ST6265 Microcontroller

J. NICOLAI, T. CASTAGNET

INTRODUCTION

Variable speed drives have become widespread in the home appliance field. The most classical way to achieve it is to use an AC universal motor with triac and a specific circuit which controls the triac firing angle. We chose rather to use a permanent magnet DC motor driven by a chopper. The chopper is driven by a high frequency PWM signal. Controlling the PWM duty cycle is equivalent to controlling the motor voltage, which in turn adjusts directly the motor speed.

Why did we make such a choice?

- The main reason is the need of a speed sensor and a closed loop regulation for the universal motor. For the permanent magnet DC motor, the speed is almost proportional to the motor supply voltage. Therefore it is sufficient to control the motor voltage (through duty cycle adjustment) if a 10% speed accuracy is sufficient. This saves the expensive speed sensor.
- The DC motor with chopper has higher efficiency, as there are no 50Hz copper and iron losses, due to lower current ripple.
- The DC operation allows to reduce the acoustic noise, especially at 100Hz.

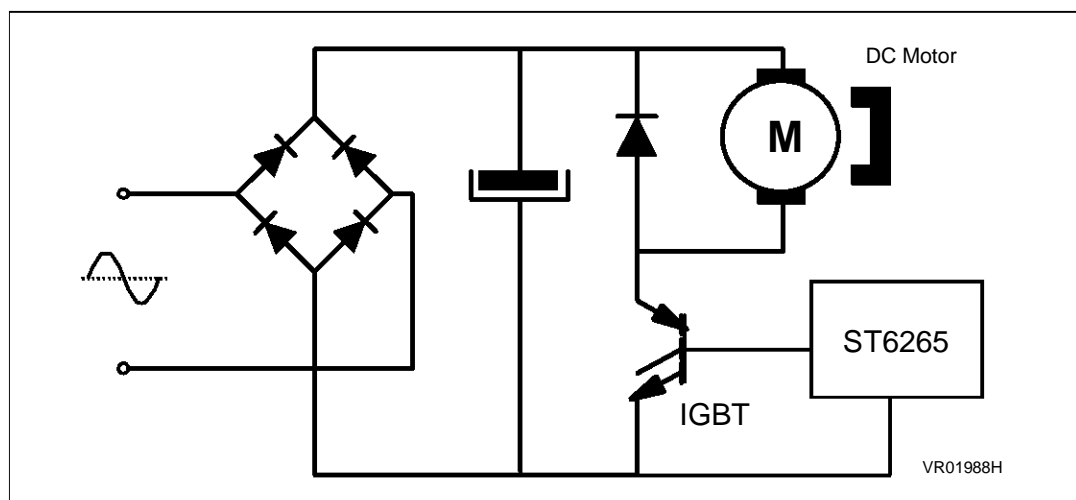
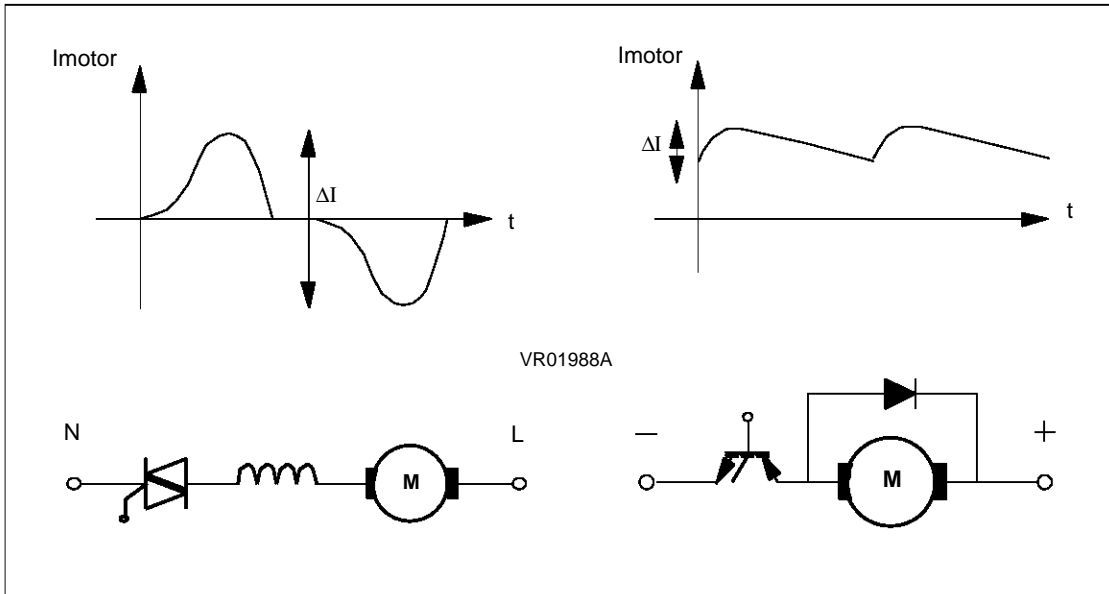
Figure 1A. Block Diagram

Figure 1B. Ac Universal Motor Versus Permanent Magnet Dc Motor: The Dc Motor Has A Lower Current Ripple And Therefore Higher Efficiency And Less



1 PRINCIPLE

The drive is supplied from the rectified mains voltage, and consists of a chopper driven by a PWM signal generated by the ST6265 Microcontroller. The motor voltage control is achieved by measuring the rectified mains voltage with the analog-to-digital converter present on the ST6265 MCU and adjusting the PWM signal duty cycle accordingly: the voltage seen by the motor is the product of the supply voltage by the PWM duty cycle. So the motor speed can be stable even if the mains voltage ripple is high, as the MCU corrects the duty cycle according to this ripple.

This avoids the use of a speed measurement device and a speed regulation loop. If this voltage measurement / duty adjustment process is carried out often enough, it also results in reducing motor current ripple, and therefore copper and motor iron losses, thus improving motor efficiency. The ripple period being 10 milli-seconds (for 50 Hz mains), the later process must occur at least every milli-second to get a large ripple reduction. In our example, the voltage compensation loop lasts 380 micro-seconds.

This drive also implements a power limitation: the motor current is measured by the analog-to-digital converter of the ST6265 MCU, and is used, together with the DC mains voltage, to compute the electrical power taken up by the motor. If it exceeds the maximum programmed power, the PWM signal duty cycle is reduced in real time by the ST6265 to stay below this maximum. This limitation requires only a few discrete components used to measure the current (2 resistors, a diode and a capacitor make up a peak current detector), it is otherwise fully software. Our application features a 300 watts maximum power, but this value can be easily adjusted by modifying a data table in the program, therefore allowing a very short design modification time if the drive must be used with various motors.

2 WHY USE a MICROCONTROLLER?

The classical implementation of such a high frequency chopper usually involves an application-specific circuit, and some analog circuitry to implement the additional features such as voltage compensation, power limitation or other.

At first look, a microcontroller should reduce the overall cost of the electronics, as a single chip would ideally replace several chips. The real world is not so simple: the ST6265 outputs the PWM signal with a 5 volts level, so a buffer is needed to drive the IGBT with 15 volts, which takes 2 or 3 signal transistors. The MCU must be supplied with a voltage between 3 and 6 volts, and the 15 volts must be generated, so we need a small auxiliary supply to extract 5 and 15 volts from the mains. In addition, the MCU cannot generally replace the fast transistor protection needed for the IGBT, as its response time is in the range of 10 micro-seconds, while less than 1 micro-second is required.

In conclusion, the MCU approach comes out at the same price, or slightly more expensive than the analog one, at least if we only account for components costs. We must, however, consider the overall system cost, not only the components cost of one function.

Three factors at least should be considered:

1- In many cases, the appliance driven by the motor includes some kind of user interface: buttons, potentiometers, LED displays, even LCD displays. If the analog approach is used, more circuitry will be needed to interface them with the motor drive. With the ST6265 MCU approach, the MCU itself can sense the buttons, potentiometers, drive LEDs...

In some existing appliances, the analog approach is used in conjunction with a microcontroller, the latter being dedicated to the user interface. This is obviously a waste, as one ST62 MCU could take care of both tasks. If we consider the case of LCD displays, which are becoming more and more popular, a dedicated LCD driver circuit is generally used to interface them to a micro.

We have implemented a software LCD drive function in the ST6265 MCU, using its free output pins; the ST6265 MCU can at the same time run the motor, drive the LCD and sense the user speed demand (in our case, two buttons "faster" and "slower"). (Cf. ref. [4] page 12).

2- Time-to-market and design time must be considered: the ST6265 MCU software contains all the characteristics of the application; it is much faster, if the application must be modified, to modify software than hardware especially when OTP parts are used for volume production.

We have implemented a software motor power limitation: the motor characteristics are known by the software in the form of a look up table. If the application must accommodate a different motor type, it is very quick to modify the data table in the software. Most modifications of the application can be achieved by simple software modifications.

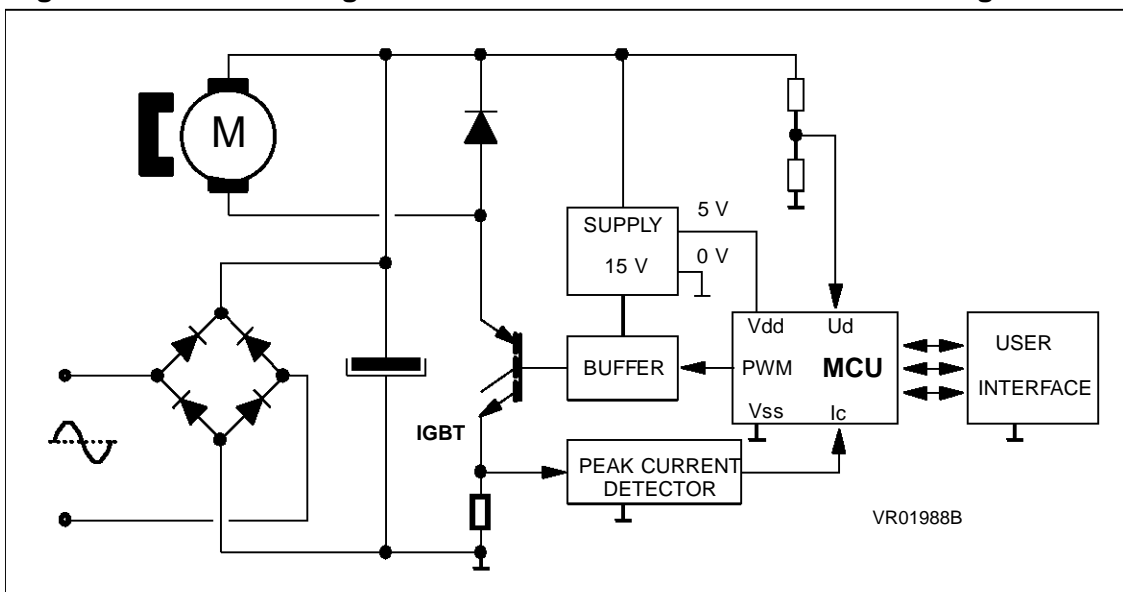
If the ST6265 MCU replaces an ASIC, the time and money saved when it is necessary to modify the initial design goes without saying. The ST62 MCU can also be used to customize the application: to limit the stock inventory, several applications, for example a low-end and high-end vacuum cleaner can use the same electronics board, with the same MCU, but the MCUs have different softwares, or the software is programmed into the ST62T65 MCU (OTP version of ST6265) at the end of the board assembly line.

3- The MCU lets you do things that are not possible otherwise. Simple examples are:

- Execute pre-programmed sequences of motor speeds, diagnostic faults and alert the user (with temperature sensors, speed sensors, ... or sometimes without them, by deducing motor speed from current, or temperature from cumulated current ...).
- Store in non-volatile memory (EEPROM) user settings which must be saved when power is turned off.

3 APPLICATION DESCRIPTION

Figure 2. Permanent Magnet Dc Motor Control With St6265 Mcu And Igbt.



The power part of the hardware is a bridge rectifier, an input filter capacitor, and the chopper. This converter drives the permanent magnet DC motor. The chopper switch is an IGBT.

The control part of the hardware is made of the ST6265 MCU, the user interface and the power interface. The latter is made of a low-power supply (5 volts for the MCU and 15 volts for the IGBT gate), a level shifter to interface MCU and IGBT, a peak current detector and a resistor bridge to measure the loop DC voltage, Ud.

The complete schematic with component values is shown in annex page 13 .

3.1 What Microcontroller to use?

The MCU generates a 5 volts PWM signal which drives the IGBT and permits to control the average voltage across the motor, and consequently the motor speed through duty cycle adjustment. On the MCU are also implemented software functions such as a line voltage compensation and a motor power limitation described later.

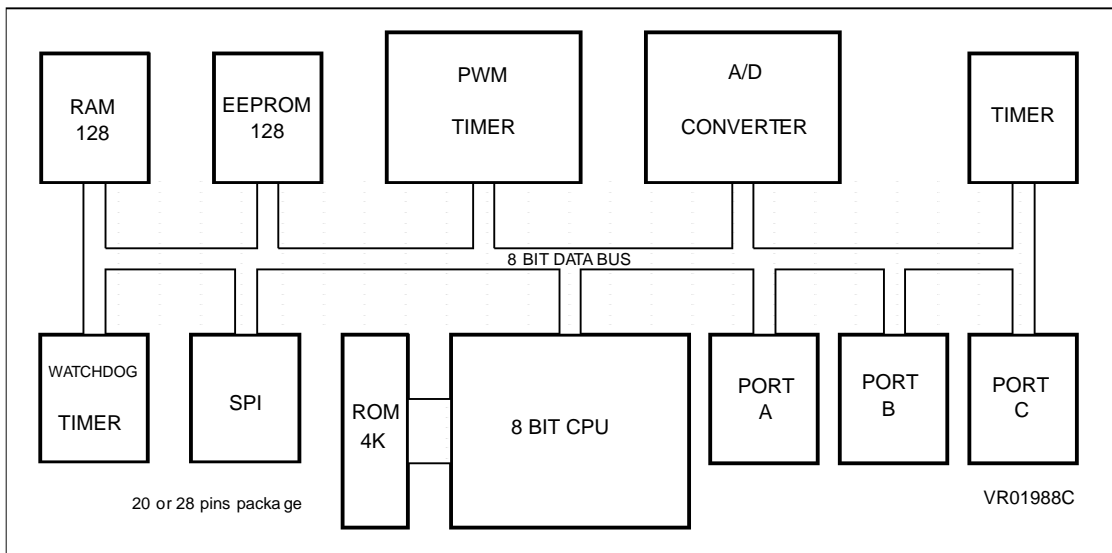
In order to implement these functions, the MCU should include the following special features:

- PWM timer: it must be autoreload to leave the CPU free for other tasks, and fast to obtain high switching frequencies.
- Analog-to-digital converter, necessary to measure the DC voltage and the motor current.
- Good noise immunity to withstand the chopper and brush sparks proximity.
- A watchdog for safe operation

We used the ST6260 / ST6265: it is a low-end 8-bits MCU with 20 pins (or 28 pins respectively), depending upon the I/O request of the user interface.

Figure 3. St6260/65 Block Diagram.

An Internal Analog-To-Digital Converter And A Fast Pwm Generator Are The



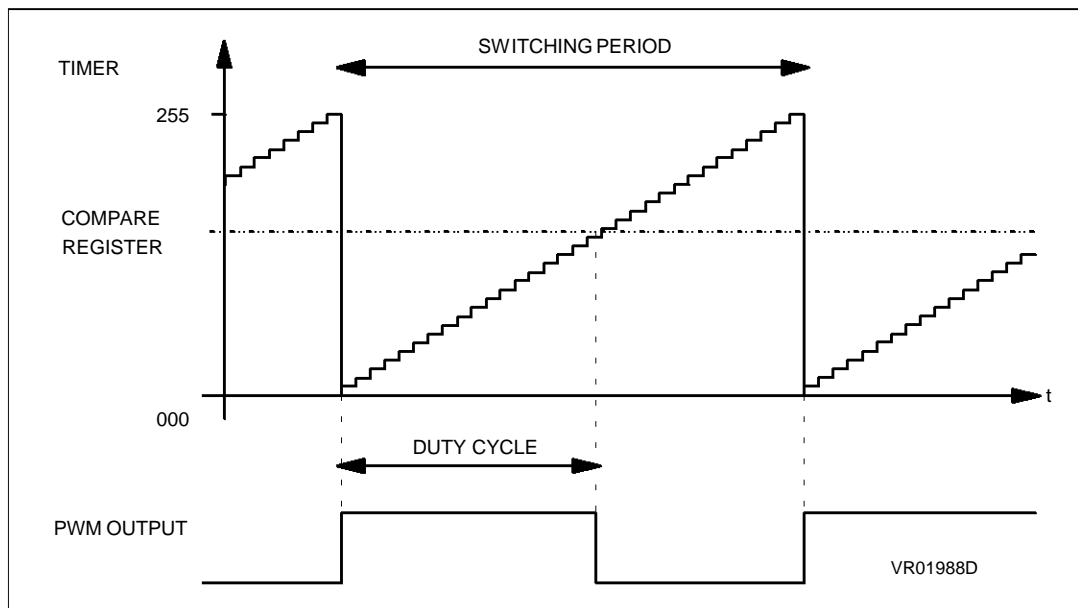
3.2 The PWM Control

The PWM feature is required to control the average voltage on the motor and therefore its speed. The autoreload timer of the ST6265 is used to achieve the PWM control (Cf. figure 4).

- At each end of count, it reloads itself without CPU operation. The CPU is then available for other tasks after it has programmed the frequency and duty cycle of the PWM signal.
- The duty cycle can be adjusted by software from 0 to 100%. The maximum frequency is: 31.25 kHz with a resolution of 256 steps (0.4% resolution).
For our application, we chose a frequency of 8 kHz.

Figure 4. Pwm Timer Operation.

Operation Of The Autoreload Pwm Timer: The Cpu Controls The Period With The Reload Register, And The Duty Cycle With The Compare Register; The



3.3 The Voltage Compensation

Using such feature on a Permanent magnet DC motor, the speed is kept almost constant by keeping the motor voltage constant even if the rectified mains voltage is strongly swinging across the filter capacitor. So a closed loop speed control is not necessary and the speed sensor is saved.

A compensation feature is implemented in order to:

- obtain a motor speed insensitive to mains voltage variations (static performance),
- reduce acoustic motor noise and dissipation by reducing the motor current ripple (dynamic performance).

The average motor voltage V_{mot} must be kept constant. The duty cycle d is modulated as a function of the direct voltage U_d around a reference point given by:

- user request duty cycle (d_0),
- nominal direct voltage (U_{dnom}).

$$\begin{aligned} V_{mot} &= d \times U_d \\ V_{mot} &= \text{constant} = d_0 \times U_{dnom} \end{aligned}$$

To achieve this function, U_d is measured using the A/D converter of the ST6265 and quantized on 32 levels, and d_0 is quantized on 16 levels. A duty cycle correction is picked up from a look-up table versus U_d and d_0 . Then the correction is added to d_0 and the sum d is loaded in the PWM timer register.

The table could also contain values chosen specifically for the application.

In order to reduce motor current ripple, the control loop must be fast, so as to minimize the time between voltage reading and duty cycle updating.

Our compensation algorithm takes 380 micro-seconds, so in one U_d ripple cycle (10 milliseconds for a 50Hz line), the correction is made approximately 26 times.

Figure 5 shows results obtained for the static performance of the compensation: the motor voltage is measured while the DC input voltage varies from 100 to 400 volts. It is shown that the motor voltage remains within a 5 to 15% range of the voltage demand. This compensation is good enough for most home appliance applications.

Figure 5. Voltage Compensation Static Performances.

The Motor Voltage (And Therefore, Motor Speed) Remains Constant While The Dc Line Voltage U_d Varies. So The Motor Voltage And The Speed

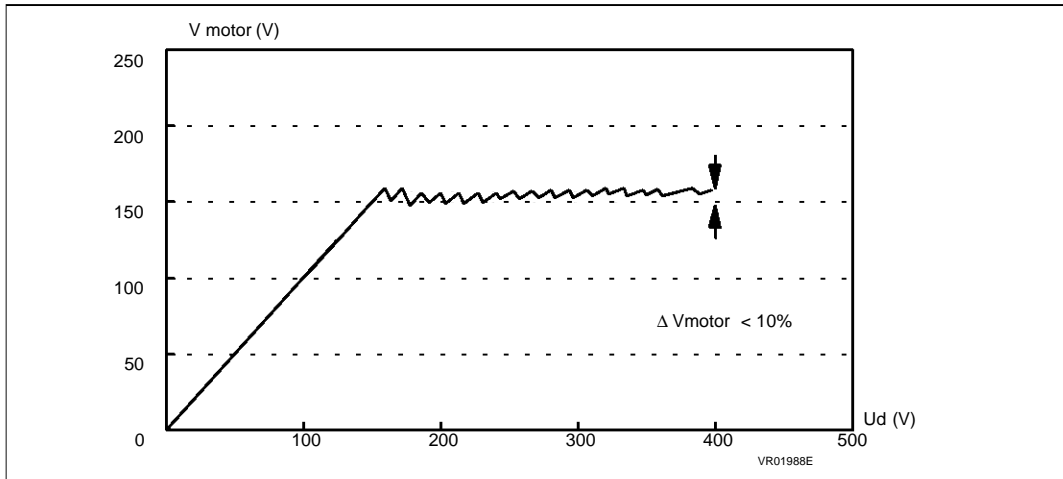
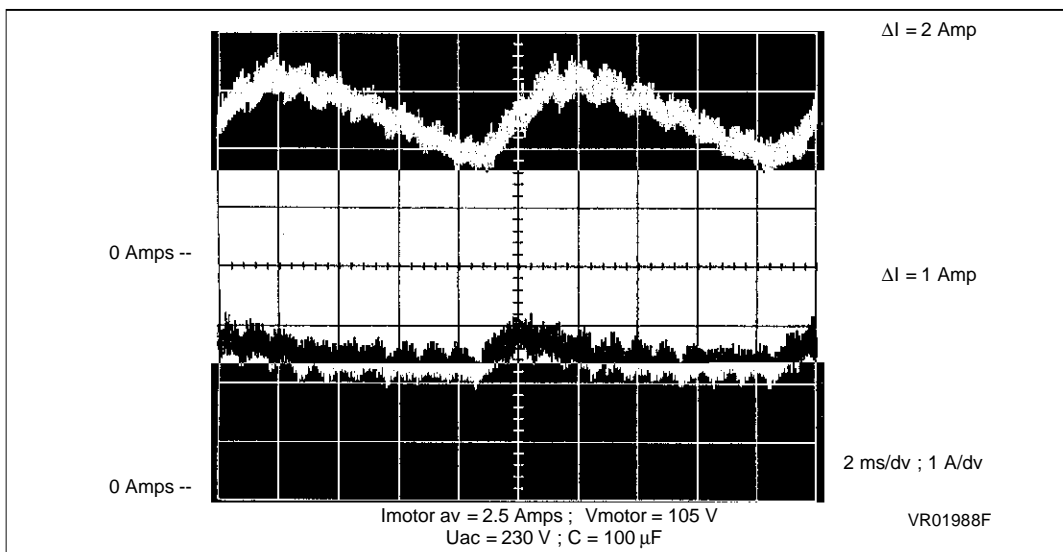


Figure 6 shows the dynamic performance of the compensation: the 100 Hz ripple in motor current is considerably reduced when using the compensation (lower trace), compared to the ripple without compensation (upper trace). The measurement is made with a loaded motor: $I_{motor} = 2.5$ amps (average) / $V_{motor} = 105$ volts. The mains voltage is 230 volts.

Figure 6. Voltage Compensation Dynamic Performance: Motor Current Ripple Reduction (1A/Div, 2Ms/Div). The Top Trace Shows The Motor Current Without Compensation. The Bottom Trace Shows A Significant



3.4 The Motor Power Limitation

The purpose of the power limitation is to keep the motor within its safe operating area in case of excessive loading (avoid overheating and hard brush switching). The motor current is evaluated every 3 milli-seconds by measuring the peak chopped current in the IGBT emitter.

If we call d the duty cycle of the PWM signal, U_d the motor DC supply voltage and I_{mot} the average current in the motor, the electrical power fed to the motor is:

$$P = V_{mot} \times I_{mot} = d \times U_d \times I_{mot}$$

If we want to limit P to a maximum value P_{max} (300 W in our example), we must keep the duty cycle below a maximum duty cycle " d_{max} ":

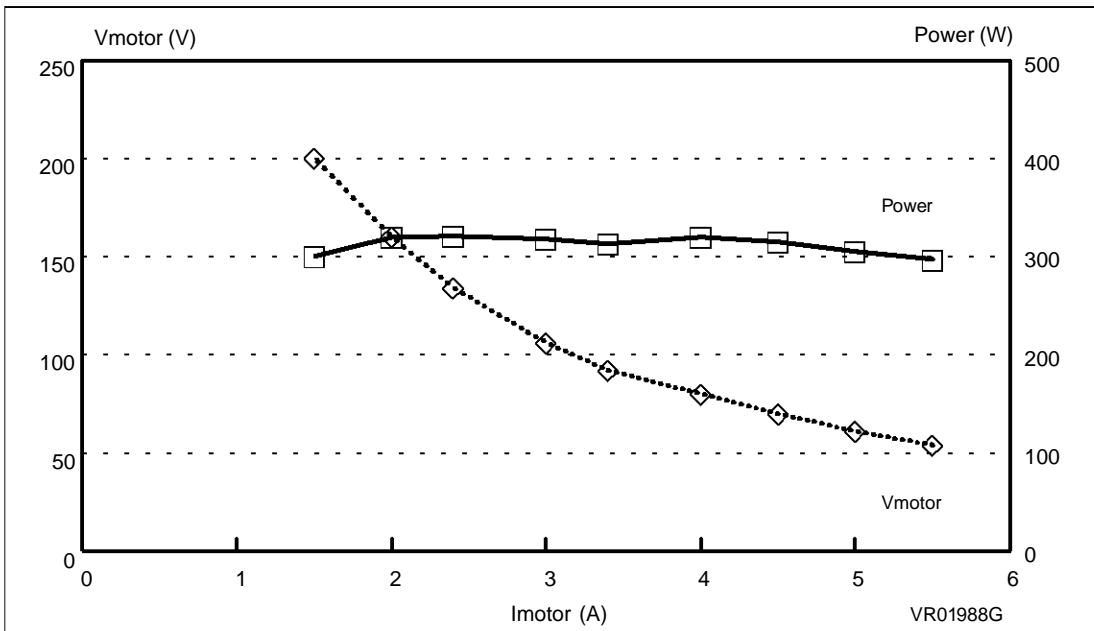
$$d \leq d_{max} \quad \text{with} \quad d_{max} = \frac{P_{max}}{U_d \times I_{mot}}$$

The program repetitively measures I_{mot} and U_d using the A/D converter of the ST6265. Then it extracts d_{max} from a table addressed by U_d and I_{mot} . It then compares d to d_{max} and decreases d if it is larger than d_{max} .

The power limitation presents "smooth" transitions between "limited" condition and "non limited" condition, in order to avoid oscillations or jerks. Figure 7 shows the measured performance of the power limitation. To modify the power limit target, the look-up table can easily be modified.

Figure 7. Motor Power Limitation Performance.

When The Motor Load Changes, The Motor Current Varies. The St6265 Mcu Measures This Current And Limits The Motor Voltage If Necessary. Software-Limited Motor Voltage (Left Scale) And Electrical Power (Right Scale)



3.5 Some Program Highlights

The total program takes 400 bytes of memory space, plus 1024 bytes for the tables. One timer remains free, and 2500 bytes remain available on the ST6260 or ST6265 for additional software or enlargement of the look-up tables.

It is possible, for example, to drive an LCD display by software, to execute pre-programmed sequences of motor speeds, current limitations, ... The EEPROM also remains available, for example to store user adjustments which must be saved when the power is off.

The line voltage compensation algorithm must be executed at high speed, as already explained, while the power limitation algorithm can be executed less often. Therefore, the complete program executes seven line compensations for each power limitation. The line voltage is compensated every 380 micro-seconds while the power is sensed and limited only every 3 milliseconds.

4 CONCLUSION

Microcontrollers are often limited to controlling and monitoring the user interfaces. This paper describes a digital solution based on a general purpose MCU with on-chip PWM generator, the ST6265. Besides the user interface such as an LCD drive, the MCU controls directly a DC motor with few external components and sophisticated control. A simple instruction set and the use of look-up tables allow for a short time-to-market, while avoiding the investment into an ASIC circuit. The demonstrated concept is applicable to home appliances (food processor, washing machine, refrigerator, HVAC ...) and to industrial applications where high speed accuracy is not required (pumps, fans ...).

5 REFERENCES

- [1] - Microcontroller and triac on 120/230v mains
A.N 392/A - Ph. RABIER and L. PERIER (SGS-THOMSON Microelectronics)

- [2] - Design with microcontrollers in noisy environment
A.N 435 - L. PERIER (SGS-THOMSON Microelectronics)

- [3] - Versatile and cost-effective induction motor drive with digital three phase generation
A.N 424 - B. MAURICE, JM. BOURGEOIS, B. SABY (SGS-THOMSON Microelectronics)

- [4] - Direct Software LCD Drive with ST621x and ST626x
A.N 594 - T. CASTAGNET, J. NICOLAI, N. MICHEL (SGS-THOMSON Microelectronics)

NOTES :

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of SGS-THOMSON Microelectronics.

© 1994 SGS-THOMSON Microelectronics - All Rights Reserved

Purchase of I²C Components by SGS-THOMSON Microelectronics, conveys a license under the Philips I²C Patent. Rights to use these components in an I²C system, is granted provided that the system conforms to the I²C Standard Specifications as defined by Philips.

SGS-THOMSON Microelectronics GROUP OF COMPANIES

Australia - Brazil - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco
The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom -
U.S.A.