SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

VERSATILE AND COST EFFECTIVE INDUCTION MOTOR DRIVE WITH DIGITAL THREE PHASE GENERATION

B. Maurice/JM. Bourgeois/B. Saby

INTRODUCTION

The three phase induction motor is a simple design, rugged, maintenance-free which appears in home appliances requiring cost effective solutions. For speed control of these motors, a frequency variation of the inverter output voltage is required. The voltage/frequency ratio must be maintained constant, so control of these motors normally require complex control circuitry for the generation of the balanced three phase sine wave outputs.

Usually the generation of the three phase PWM signals may be controlled by a dedicated circuit, such as the SGS-THOMSON L6234, which is driven by a separate microcontroller. This solution is optimum while performance prevails over cost.

The solution demonstrated in this application note is a simplified solution using a standard ST9 microcontroller which includes large on-chip ROM memory and an internal Direct Memory Access (DMA) controller. This combination reduces the need of dedicated ICs (hardware being replaced by software), and allows over 50% of the CPU time to perform control, environmental and supervision tasks.

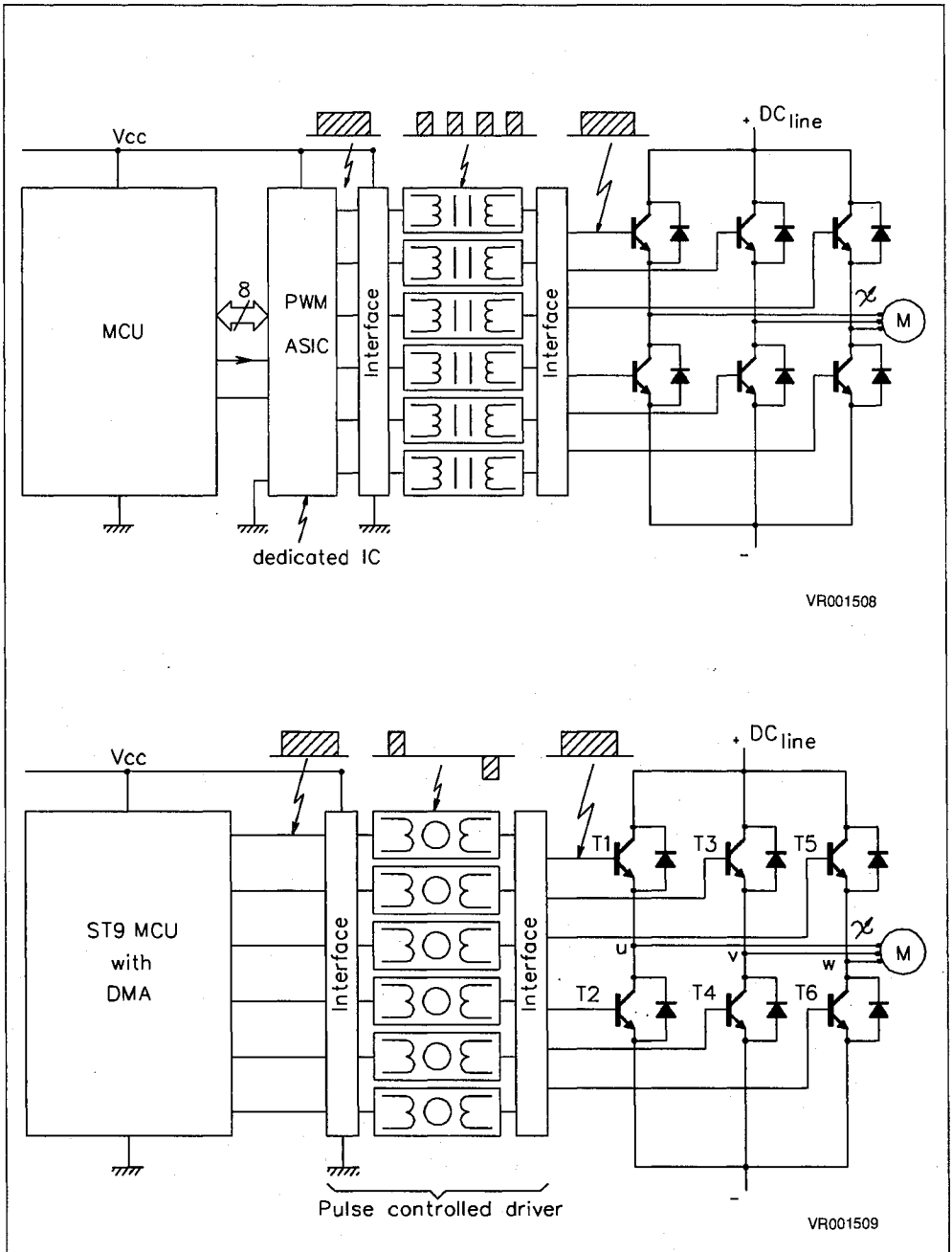
A practical solution to quantize three phase sine-waves, and to create the corresponding DMA table is shown allowing motor voltage and motor frequency to be chosen independently. A dead time avoiding cross conduction through the bridge is also created by software. Very low acoustic noise operation can be achieved despite a switching frequency below 10kHz, due to a shifting of the switching instants leading to a virtual doubling of switching frequency.

Each of the six digital outputs of the ST9 sets directly the state of the six power MOSFETs (or IGBTs) of the bridge via an insulated interface. This interface is described in the second part of this note. The fully isolated pulse controlled gate driver requires no floating auxiliary supply, meets safety standards and achieves a large dV/dt immunity.

Figure 1 shows how to generate a three phase sine wave by modulation of pulse width. This modulation is often obtained with a special dedicated IC controlled by a MCU (above).

Using a MCU having large memory integrated on the chip combined with DMA, spares the use of dedicated IC (below). Hardware is replaced by software. The sine waves are directly synthesized by the MCU.

Figure 1. Three Phase PWM Generation Techniques



DIGITAL CONTROL OF POWER SWITCHES

In this proposed solution, the ST9 microcontroller controls simultaneously the ON- and OFF- states of the six power switches of the inverter bridge. All these instantaneous ON-OFF states are stored in internal memory (ROM) and are sequentially transferred (every 5s for example) to six bits of a parallel output port by DMA (see Figure 2). The voltage level 0-5V of each output bit drives directly the gate interfaces of the six power switches .

All data corresponding to the switching duty cycle values is permanently stored in ROM and generates the quantized three phase sine waves. A dead time between adjacent Power switches is also stored, avoiding cross-conduction through the power bridge. The motor frequency and motor voltage are also stored independently.

The major part of the ROM is occupied by this permanent data, used to generate, step by step, the three phase sine-waves. This data is grouped in several tables (patterns), constituting series of bytes that have to be sequentially output on the parallel output port. A full scrolling of each pattern corresponds to a complete switching basic cycle of the six power switches. This is repeated the necessary number of times to complete the step duration of sine wave. The following pattern will then be scrolled to realize the following step.

This direct sequential transfer from memory to output port is performed by DMA [2], and is self operating. The central unit only works when the last byte of one pattern appears, the program then determines whether the same pattern must be scrolled again, or if another new pattern has to be scrolled.

All patterns needed for an application, as well as the program managing their scrolling order and their number of repetitions, are to be created and stored in ROM.

MOTOR DRIVE CONFIGURATION

Microcontroller

The ST9036 microcontroller from the ST9 family with 16k-byte of ROM or EPROM memory [1], of which only one output port and one multifunction timer are used for PWM generation. Six bits of its output port are gathered in pairs, one pair for every bridge leg (phases: u,v,w). The two bits remaining free can be used, for example, either to control two other power switches (i.e for heat control in a washer), or to generate a synchronized signal to perform measurement of V/I phase.

The ST9 microcontroller is able to manage two further functions:

a) Slow operations for motor and environment controls, such as timing of sequential operations, speed control, safety supervision tasks, etc. (These are not detailed in this application note).

b) Faster operations for real time management of the states of the power switches for PWM generation.

All others functionalities of the ST9036 remain available, such as other I/O ports, Timers, Analog/Digital converters and all interrupt functions.

Figure 2. DMA Transfer to control power switches

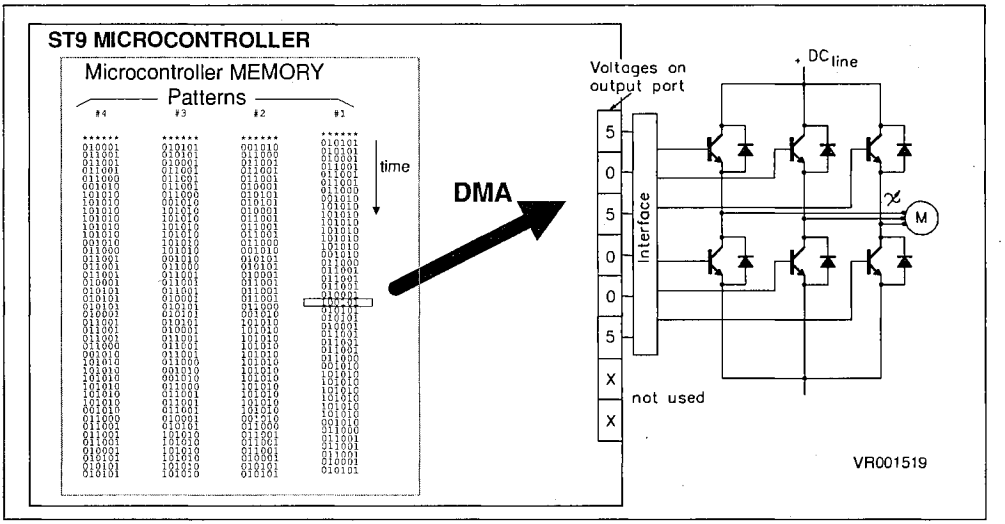
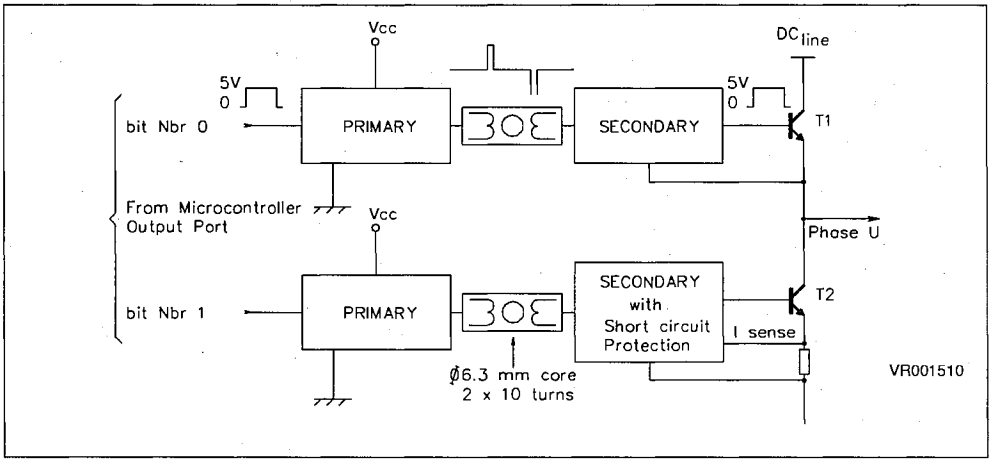


Figure 3. Driver for one Bridge Leg



In the practical example described in the following sections, ST9 is not heavily occupied by these real time operations:

- Using DMA is similar to slowing down the ST9 and engages only 35-40% of the CPU time.
- Speed control (frequency variation) needs only few instruction lines but no memory space. The memory space is mainly used to store necessary data to generate six various three phase voltages supplying the motor (1k-byte for each voltage).

Drivers For Power Switches

The driver interfaces the ST9 output port to the gate of the power switches.

- it converts the output level (5V) to the required gate-source voltage level (15V) of IGBT or Power-MOSFET.
- it provides a galvanic isolation.
- it protects against current surges and short circuits.

It is constituted by six independent circuits for the six power switches. Each is a pulse controlled driver [4] including: (see Figure 3)

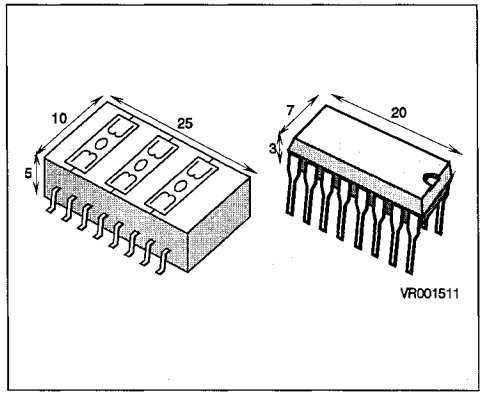
- a primary circuit to create a calibrated Pulse with short duration.
- a small pulse transformer. (DIL molded package)
- a floating secondary circuit operating without any auxiliary supply and including the autonomous short circuit protection.

The primary circuit differentiates the logic level input signal. The positive and negative calibrated output pulses (±15V/0.5µs) correspond to the switch-on/switch-off command. The primary circuit output stage is a full bridge having a low output impedance in order to obtain short rise times and high amplitude current pulses.

The pulse transformer can be small. A ferrite core of 6.3mm diameter with 10 turns is sufficient as it has to sustain 15V for 0.5µs. In this application, three core transformers are housed in the same standard or SMD package [3].

The secondary circuit needs no supply and uses the input gate capacitor of the Power-MOSFET or IGBT like an R/S memory.latch. The required energy is limited to charge and discharge the input gate capacitor. During the OFF-state, a low impedance is maintained across the gate-source of the Power switch, avoiding any reconduction due to externally applied dV/dt.

Figure 4. Transformer Core Size vs 16 pin DIL



In several applications, when isolation between the power and control sections is not mandatory, the low side driver can be a simple non-insulated driver. Nevertheless, the fully isolated solution performs high dV/dt immunity and meets insulation standards.

DC/AC Inverter

For this function, a three-phase bridge with six switches (Power-MOSFETs or IGBTs) is used. (Figure 1). The two switches of each bridge leg are opposite phase controlled. A dead time, avoiding simultaneous conduction, is generated directly by the ST9036 microcontroller.

Sine wave generation: (Figure 5)

The voltage on middle point of "u-phase" bridge leg is given by:

$V_u = V_{DC}.\delta_u$ $\quad \delta_u$ = u-phase duty cycle

$t_{on\ hi}$ = "ON state" duration of high side switch

$\delta = t_{on\ hi}/T_s$ $\quad T_s$ = switching period

$V_{uw} = V_u - V_w$ $\quad V_{uw}$ = phase to phase motor voltage

If δ_u is sinusoidal modulated, the average voltage on half bridge middle point describes sinusoidal wave form centered to $V_{DC}/2$. To avoid DC components in the motor, each phase voltage has to be symmetrical compared to $V_{DC}/2$.

Motor voltage value

Motor voltage is maximal when the duty cycle modulation varies from 0% to 100% (modulation depth: K=100%)

Motor voltage is minimal (nil) when modulation depth K=0; δ does not vary and is equal to 50% (Figure 5b)

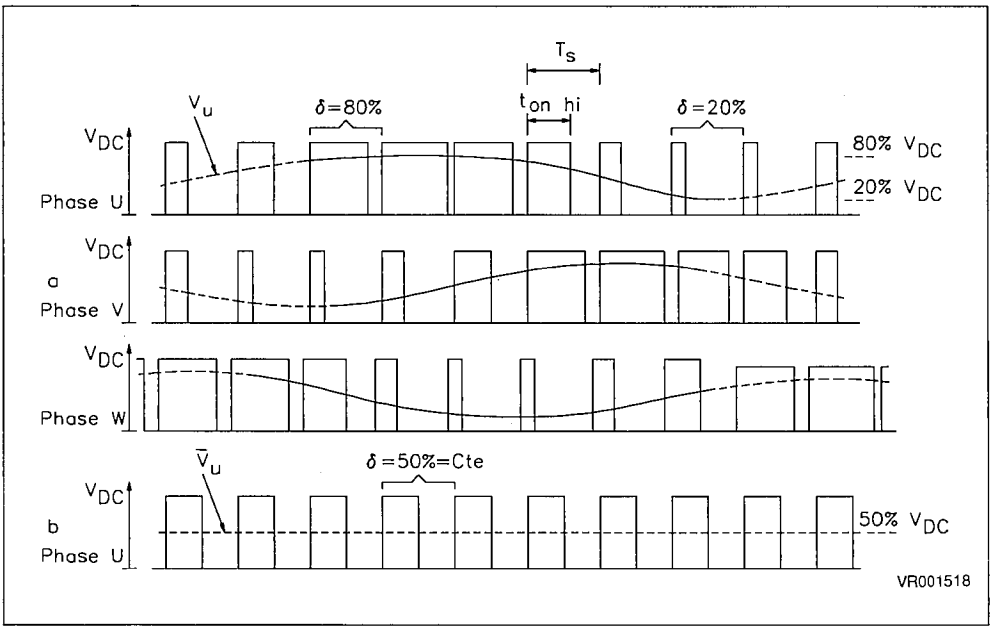
Sine wave frequency variation

This is obtained by varying the frequency of the duty cycle modulation.

CREATING TABLES OF DATA

The variable speed drive of induction motors requires generating three voltage sine waves and control of their amplitude, phase and frequency. The first step is to digitize the three phase system in order to create all the necessary data to be stored into the ROM of the ST9 microcontroller.

Figure 5. Sine Wave Generation at the output of one bridge leg
a. Modulation depth 60%, duty cycle 20 to 80% **b. Constant duty cycle**



Fundamental period quantification

The fundamental period of motor voltage is divided into 24 "segments"; (each segment equals 15° of arc). This gives a good sine wave accuracy in many applications. During each "segment" the voltage is a percentage of the DC line voltage, given by duty cycle (δ). For example, the duty cycle must be 55% during the segment from 165° to 180° for phase U (Figure 6).

Creating the duty cycle table

The second step is to establish a table giving, for each segment, the duty cycle value (δ) for each of the three phases. In fact δ1, δ3, δ5 are duty cycle values for each high-side switch (T1, T3, T5). The low side switches are in the opposite states and their duty cycle value is complementary to 100%. This entire table defines exactly the three-phase sine wave system during one period (T_O) and for one motor voltage. (Figure 7) These table values respect phase balance and avoid neutral currents. To achieve these conditions it must be ensured that:

a) on each line, the sum of the three duty cycle values is constant (equal to 150%).

b) The duty cycle has a symmetrical value either side of 50%. In practice the quantized values have to be chosen close to the mathematical value of sinus for only a quarter of the period, then symmetrically repeated respecting the condition (a).

This duty cycle table is not stored in ROM. It only defines the necessary data to create the patterns. One line of this table defines one pattern (see following section).

Figure 6. The fundamental period divided into "segments"

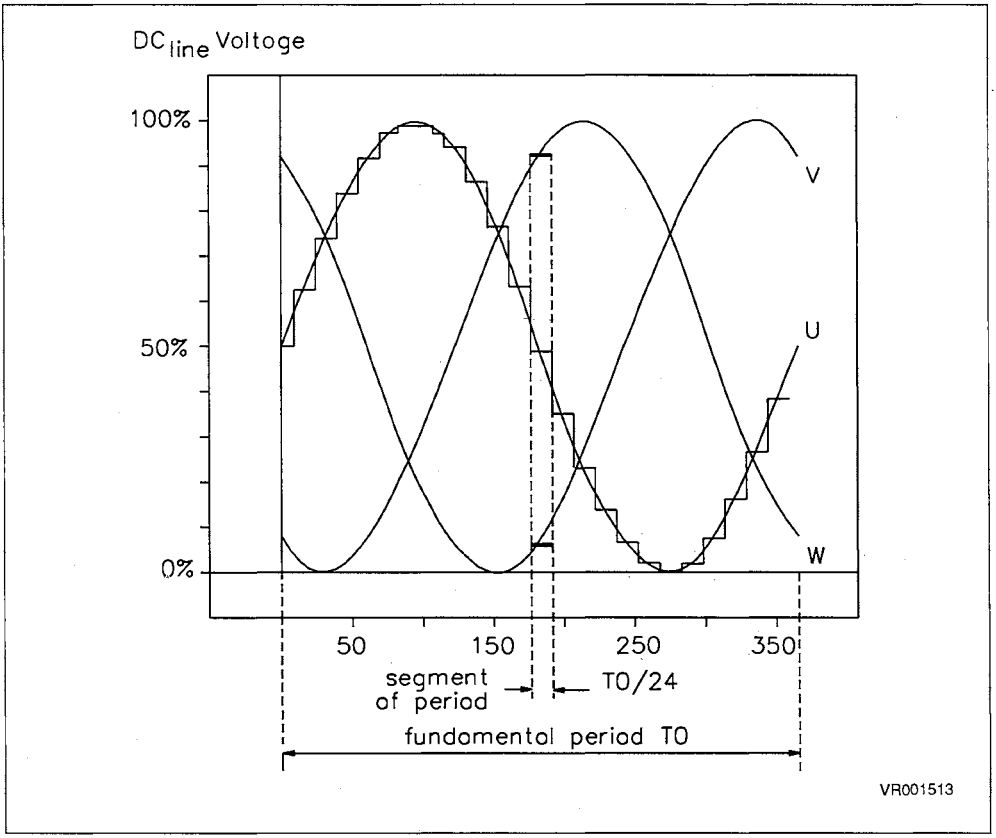
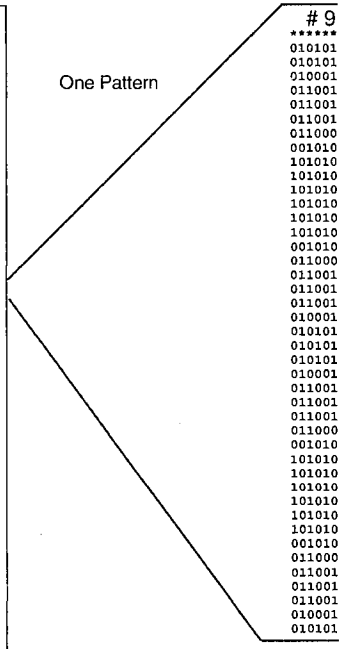


Figure 7 . Duty cycle table defining data to create patterns $V_{PHASE} = 0.6xV_{LINE}$

Pattern #	U δ1%	V δ3%	W δ5%
1	80	40	30
2	80	45	25
3	75	55	20
4	70	60	20
5	60	70	20
6	55	75	20
7	45	80	25
8	40	80	30
9	30	80	40
10	25	80	45
11	20	75	55
12	20	70	60
13	20	60	70
14	20	55	75
15	25	45	80
16	30	40	80
17	40	30	80
18	45	25	80
19	55	20	75
20	60	20	70
21	70	20	60
22	75	20	55
23	80	25	45
24	80	30	40



Pattern definition

A pattern is a succession of bytes stored in memory. Each bit (1;0) of these bytes gives the instantaneous state (ON;OFF) of each of the 6 six power switches (Figures 7&9). Pattern contains number of bytes necessary to define one entire basic switching cycle.

A particular pattern has to be created for each segment of the sine wave period. All these patterns are stored in the ST9036 ROM.

For example (Figure 8), a pattern contains sequence of 42 bytes defining one basic switching cycle. The switching period T_s, shared into 42 units of times, gives a good sensibility of duty cycle adjustment of about 2.5% (1/42th). This time unit corresponds to the rhythm of the DMA timer and its duration is chosen as a multiple of the ST9 microcontroller clock period (0.25µs).

In this example, one unit of time equals 4.75µs in order to have a pattern scrolling time or switching period T_s = 200µs. This corresponds to 5kHz of switching frequency.

Two dead times (one time unit each) at every state change of adjacent switches avoid cross conduction of the bridge leg.

Figure 8. Example of switching cycle for transistor T1

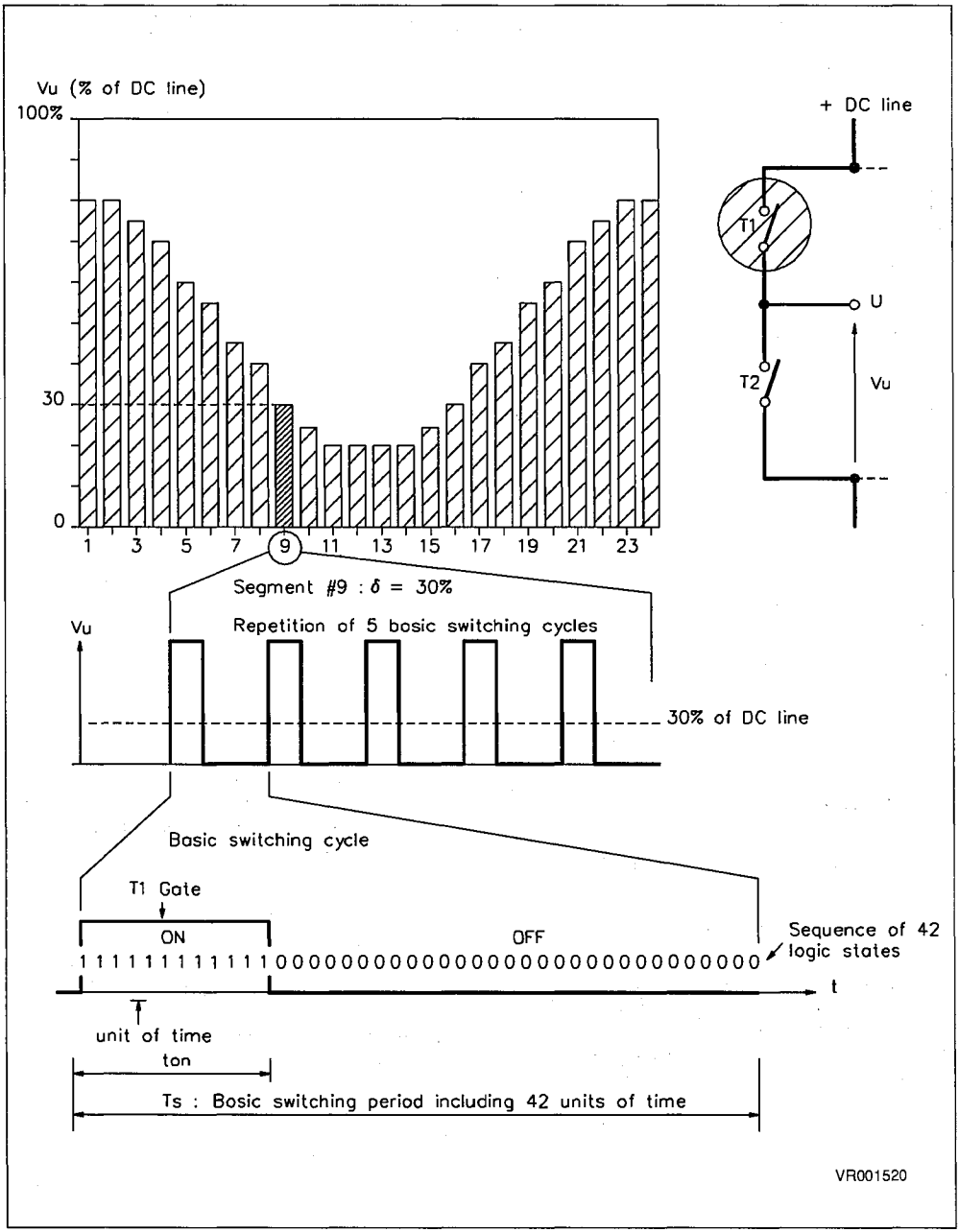


Figure 9. Pattern table

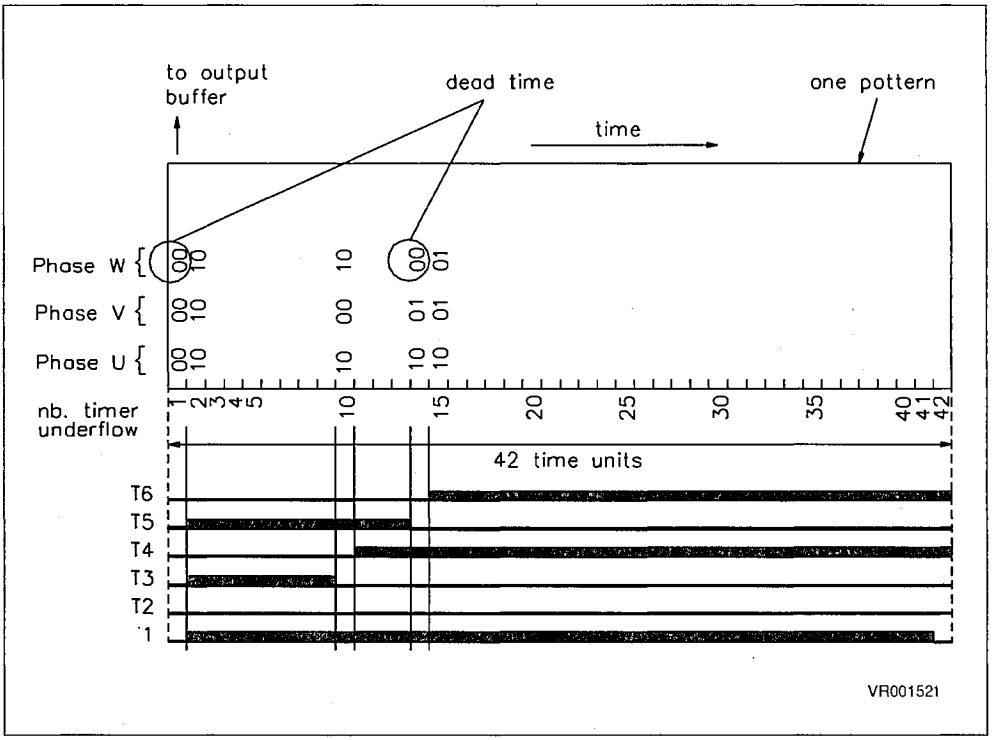


Figure 9: One pattern is an elementary table grouping all necessary bits to define the basic switching cycle of every six inverter switches. Bytes are sequentially read by DMA, and transferred to the output buffer. The resulting switching cycles shown on the bottom of the figure gives the following duty cycles:

-phase U: δ1 (T1) = 100% ; δ2 (T2) = 100% - δ1

-phase V: δ3 (T3) = 20% ; δ4 (T4) = 100% - δ3

-phase W: δ5 (T5) = 30% ; δ6 (T6) = 100% - δ5

MOTOR VOLTAGE CONTROL

One table of duty cycle defines the 24 stored patterns (set of patterns) containing information to one AC motor voltage. As an example, the peak value of phase voltage generated by the table given on Figure 7 is equal to 60% of V_{DC} line. It is necessary to create a set of patterns for each of the needed motor voltage.

The motor voltage can be controlled independently of the frequency. This voltage depends on the set of pattern which the DMA is reading. By storing within ROM and reading different set of patterns, the voltage across the motor can be changed and shaped. In this example, a set of patterns includes 24 patterns of 42 bytes each = 1008 bytes.

Figure 10 shows δ1 duty cycle of the T1 switch for various values of motor voltage. On the right side the chart gives the corresponding values of δ3 and δ5. Each line of these charts defines a pattern. These values respect the phase balance and avoid current in neutral line. The useful RMS voltage across motor phases is given by:

$$V_{RMS} = \frac{1}{2}.K.\sqrt{\frac{3}{2}}.V_{DCline}$$

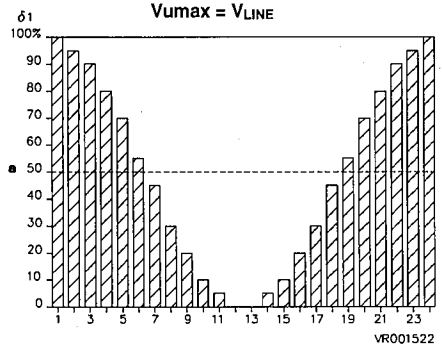
The K factor corresponds to modulation depth of the duty cycle (δ) as shown on examples (a,b,c).

a: K= 1.0 when duty cycle varies from 0% to 100%

b: K= 0.6 when duty cycle varies from 20% to 80%

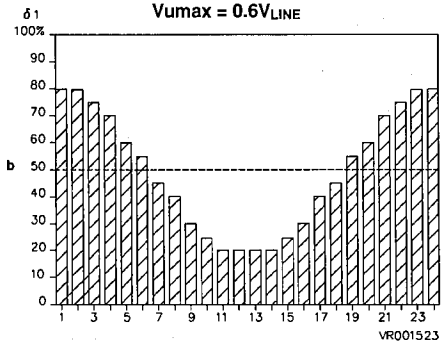
c: K= 0.2 when duty cycle varies from 40% to 60%

Figure 10. δ1 Duty Cycle of T1 Switch



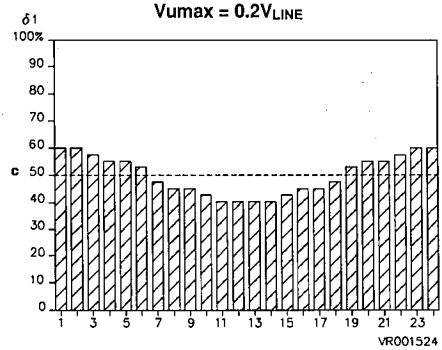
Set of Patterns

Pattern	δ1	δ3	δ5
1	100	30	20
2	95	45	10
3	90	55	5
4	80	70	0
5	70	80	0
6	55	90	5
7	45	95	10
8	30	100	20
9	20	100	30
10	10	95	45
11	5	90	55
12	0	80	70
13	0	70	80
14	5	55	90
15	10	45	95
16	20	30	100
17	30	20	100
18	45	10	95
19	55	5	90
20	70	0	80
21	80	0	70
22	90	5	55
23	95	10	45
24	100	20	30



One Pattern

1	80	40	30
2	80	45	25
3	75	55	20
4	70	60	20
5	60	70	20
6	55	75	20
7	45	80	25
8	40	80	30
9	30	80	40
10	25	80	45
11	20	75	55
12	20	70	60
13	20	60	70
14	20	55	75
15	25	45	80
16	30	40	80
17	40	30	80
18	45	25	80
19	55	20	75
20	60	20	70
21	70	20	60
22	75	20	55
23	80	25	45
24	80	30	40



Set of Patterns

1	60	45	45
2	60	48	42
3	58	52	40
4	55	55	40
5	55	55	40
6	52	58	40
7	48	60	42
8	45	60	45
9	45	60	45
10	42	60	48
11	40	58	52
12	40	55	55
13	40	55	55
14	40	52	58
15	42	48	60
16	45	45	60
17	45	45	60
18	48	42	60
19	52	40	58
20	55	40	55
21	55	40	55
22	58	40	52
23	60	42	48
24	60	45	45

MOTOR FREQUENCY CONTROL

Motor frequency is controlled via duration of the fundamental period To.

The shortest duration of the period (the highest frequency) is reached when each segment of this period corresponds to only one reading of the corresponding pattern. In our example (Figure 11) the pattern reading duration equals 200µs, and with 24 segments.

When each segment corresponds to two readings of pattern, the fundamental period is twice as long. Thus the frequency (motor speed) can be controlled step by step whether the pattern is read once or several times. Consequently when starting from the highest frequency, it is possible to have discrete submultiples of frequency.

F = Fo/N N = number of times of patterns being read

The speed resolution is low for high motor speed, but high for low motor speed. So, to perform the speed control by software, it is sufficient to give the number of times the pattern is to be read. For example, when repeating 20 times the same pattern the following results are obtained:

4.75µs = time unit

42 = number of time units per pattern

20 = number of patterns (or switching cycle) reading per segment

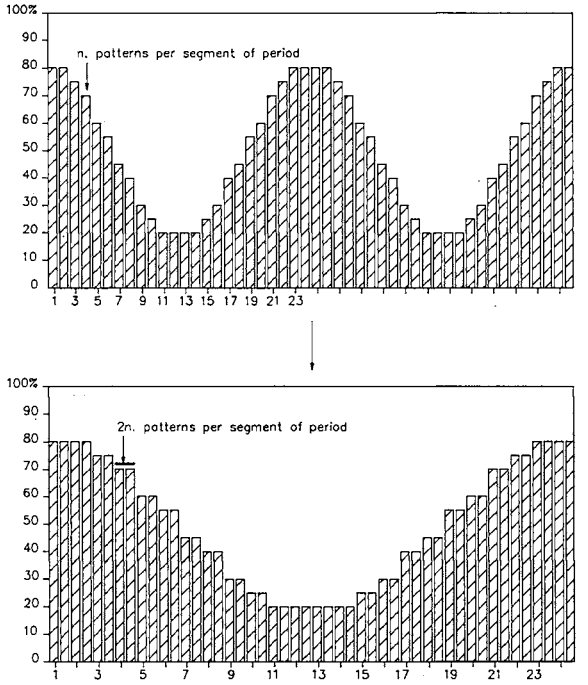
24 = number of segments per fundamental period

This gives a fundamental frequency of 10.4Hz

Another way to adjust motor speed by software is to change the DMA timer period. That is equivalent to modifying the "time unit" duration. The reduction of time unit duration is limited by the highest consumption of CPU time to be accepted and the shortest permissible dead time is according to power switches used.

By combining these two methods, pattern repetition and timer variation, it is possible to perform quasi-continuous variation of motor speed.

Figure 11. Fundamental Frequency variation
a. To = 200x24=4800 ms Fo=208Hz
b. To = 200x2x24=9600ms Fo=104Hz



VR001525

DEPHASING SWITCHING INSTANTS

When creating the pattern, the instant of switching can be chosen specifically for each bridge leg. For example it is possible to simultaneously turn-on all the high side switches (T_1, T_3, T_5) and stop them when respectively each duty cycle is reached (Figure 12a).

Through other ways for the same duty cycle the ON-state is centered at the middle of pattern (Figure 12b).

Various other possibilities can be chosen to create the pattern. The acoustic noise of the motor will depend on this choice. For example the pattern shown in Figure 12c gives a large current ripple and very noisy motor,while, on the contrary, Figure 12b gives a noiseless motor according to small current ripple (shown at 20ms/div; 2A/div).

When the three high side switches or the three low side switches are simultaneously ON or simultaneously OFF, no energy is transmitted into the motor, which is freewheeling. Another possibility is to choose simultaneously the OFF-state rather than the ON state simultaneously as shown Figure 12e to compare with Figure 12b. In this case two switches of one bridge-leg are not switched and switching losses are reduced.

Figure 12 shows how various possibilities can be chosen for pattern creation .
Current ripple and acoustic noise of motor will depend on this choice.

On the right side, motor current with:

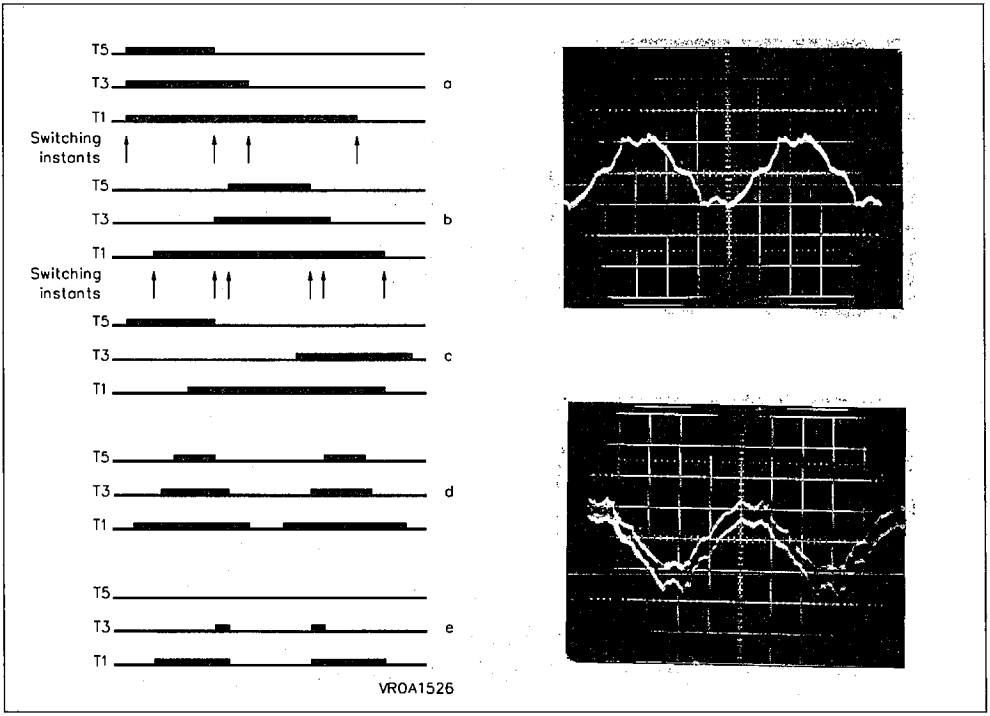
- noiseless motor according to small current ripple.
- noisy motor according to large current ripple.

As the energy is transmitted when switches are not in the same state, the rule to create a pattern is to maximize the instants where the switches are in the same state and simultaneously shift the switching instants.

All these methods can be used to obtain very low acoustic noise operation in spite of a switching frequency below 10kHz.

Without reducing the time unit, it is possible to increase the acoustic frequency by sharing in two equal parts each duty cycle time (δ=60% $\Rightarrow$ δ=30%+30%) . The switching frequency is doubled and acoustic noise is close to the inaudible region and becomes very low (Figure 12d)

Figure 12. Pattern options affecting ripple and acoustic noise



EXPERIMENTAL EXAMPLES

Figure 13 shows an example of generated three phase PWM signals on microcontroller outputs. It represents three control signals for T1, T3, T5. The set of pattern corresponds to a modulation depth of 100% as shown on table Figure 10a. The phase angle between each phase is 120%.

Figure 14 shows current measured in motor phase (20ms/div, 2A/div).
a) obtained with set of pattern shown on Figure 10b, and repeated twenty time, f=10Hz
b) patterns are repeated twice, f=100Hz.

The used set of pattern (at 60%) of Figure 10b, combined with doubly of switching instants (Figure 12d), gives a well defined sine wave.

Very little ripple of current and doubling switching frequency give a noiseless motor operation.

The motor is speeded up by repeating patterns only twice (Figure 14b). Simultaneously the motor voltage is increased by using set of pattern (Figure 10a) having modulation depth of 100%.

Figure 13. Three microcontroller outputs: 5ms/div, 5V/div

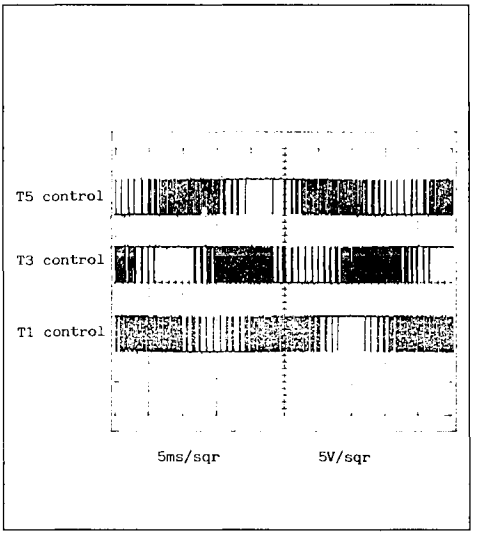
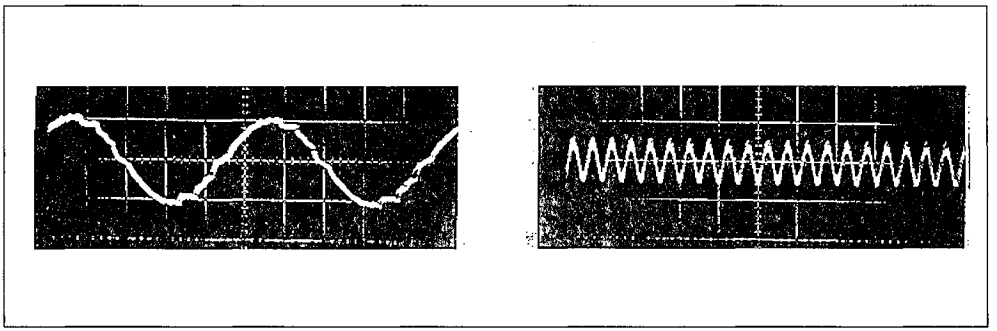


Figure 14. Current measured in motor phase



SUMMARY

For large volume applications such as washing machines, air conditioning or cooling pump motor drives, cost optimization is a key issue. The solution to drive induction motor presented in this paper simplifies conventional digital solution. Using a ST9 microcontroller with Direct Memory Access and fast data transfer, replaces dedicated ICs by software or more precisely by data stored in microcontroller memory.

The proposed solution is very versatile because a standard microcontroller, the ST9036, can be programmed for various applications, only the software will has to be adapted.

This note presents methods to generate data in order to shift the switching instants of inverter switches. This allows to reduce motor acoustic noise in spite of switching frequency being below 10 kHz, and to minimize motor and converter losses.

The described pulse controlled gate driver uses standard components and small enough core transformers that can be fitted into a Surface Mounting Package. This way offers a cheap fully galvanic insulation when required.

REFERENCES

[1].ST9 family high performance 8/16 bit MCU
SGS-THOMSON Microelectronics - Info pack

[2].3-phase motor drive using the ST9 multi-function timer and DMA
B. SABY - SGS-THOMSON Microelectronics - Application Note.

[3].Data-book of F.E.E 39270 Orgelet France

[4].New Isolated Gate Drive for Power MOSFET and IGBT
JM. BOURGEOIS - E.P.E - Firenze/I Sept 1991

[5].External DMA mode: I/O data transfer synchronized by timer
P. GUILLEMIN - SGS-THOMSON Microelectronics -Application Note AN418

[6].Environment design rules of MOSFET in medium Power application
B. MAURICE - P.C.I.M. Munich/G 1989 - Proceedings book

APPENDIX MICROCONTROLLER WITH DMA

The feature of microcontroller with DMA (direct Memory Access) consists in having a possibility of direct access between microcontroller memory and its on-chip peripherals. Moreover, one of the parallel I/O ports can be coupled with the timer's DMA channel, allowing fast data transfers between memory and this I/O port with minimum CPU overhead. Data transfers are scheduled by the timer.

The only task of the Microcontroller software is to specify which pattern is to be read by the DMA channel at a given time in order to reproduce the three-phase sinewaves, as described in the previous sections. After a complete pattern transfer, the Micro-controller CPU is interrupted (DMA End of block interrupt) and the DMA should start to read the next pattern.

In order to achieve high speed continuous transfers without stringent response time requirements for this End of block interrupt, a "swap mode" is used: while a pattern is read by the DMA channel, the subsequent pattern can be prepared in advance; so, once the last byte of the pattern is read, the DMA automatically switches to the new pattern while the old one can be updated during the DMA End of block interrupt routine.

First tests show that the DMA operation in swap mode, as described hereabove, accounts for 35-40% of the total available CPU time of the Microcontroller. Therefore, thanks to its processing power, the Microcontroller can easily perform any control and supervision task in addition to this DMA-driven PWM generation.