Sensorless Motor Drive with the ST62 MCU + TRIAC

T. CASTAGNET

INTRODUCTION

Home appliance applications are requiring more and more electronic control in order to meet new requests and constraints of consumers.

Microcontrollers have been typically limited to high end applications because their performance appears to be overrated when related to the functions of the application. In reality, home appliances require microcontrollers which trade closely between the compromise between cost and performance.

An a.c. universal motor is a cost optimized solution for home appliance applications including food processor and drill applications.
This Application Note shows that the capabilities of simple 8-bit microcontroller allows the design of cost effective speed drive controller with increased functionality. When associated to a triac these microcontrollers become key components in the design of a sensorless speed control.

1 THE CONTROL OF THE MOTOR SPEED

An a.c. universal motor is a brush motor with a serial excitation. Its stator windings are connected in series with the rotor, and its flux is proportional to the motor current. The motor torque is theoretically proportional to the square of the current, so it is always positive: the speed direction is insensitive to the current direction, and the motor can be supplied in a.c. or d.c. modes. Control of the speed is obtained by adjustment of the motor voltage. This control is achieved by a phase angle method in a.c. mode, or by a Pulse Width Modulation method in d.c. mode.

Figure 1. The triac is a key device for the a.c. drive of the universal motor.

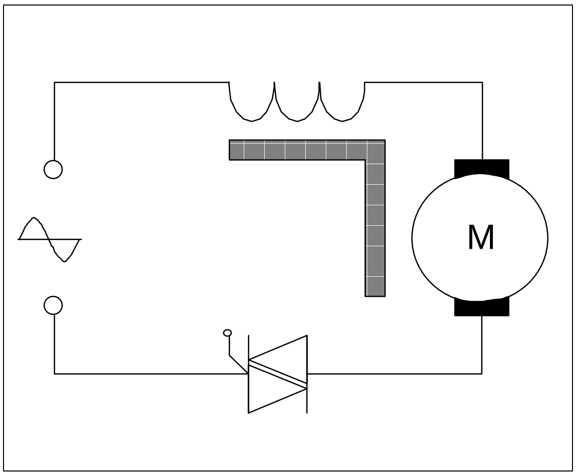
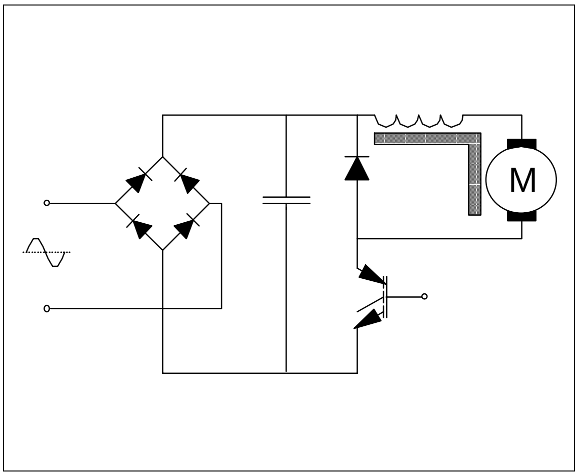


Figure 2. A d.c. drive for the universal motor: the P.W.M. chopper with IGBT.



For a fixed motor voltage the motor characteristic shows that the motor speed changes when the torque is varied: the control of the speed requires feed-back of the speed itself.

Figure 3. Universal motor torque T versus the motor current Imot.

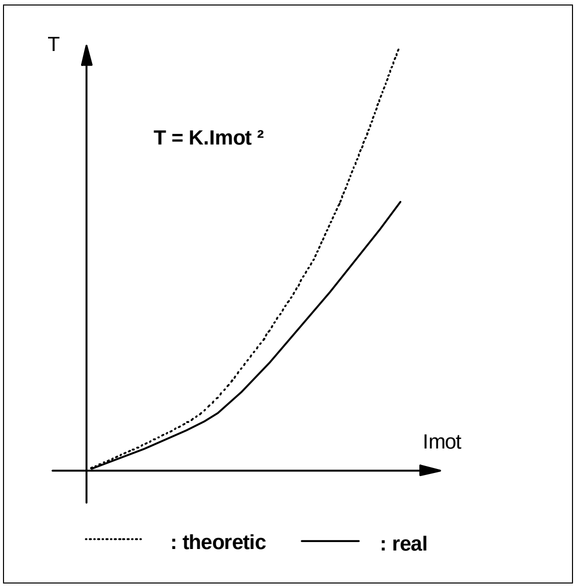
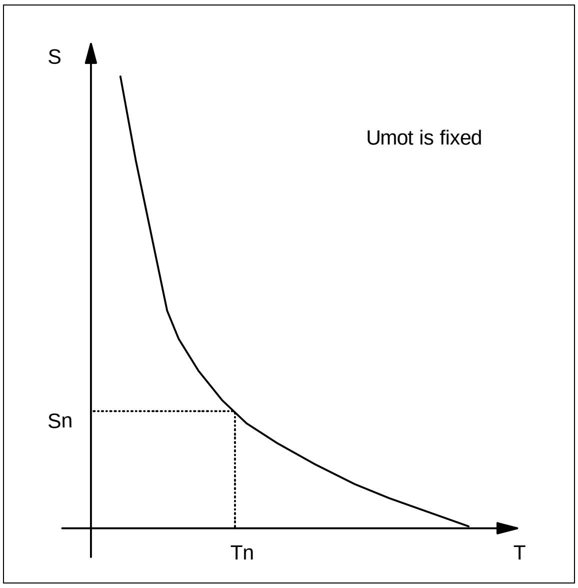


Figure 4. Universal motor characteristic speed torque (S,T) for a fixed motor voltage Umot.



The tachogenerator is a classical speed sensor solution for home appliances. When the accuracy of the speed is not a critical parameter, speed control is also possible without any external speed sensors: thus reducing the number of components in the speed drive.

Figure 5. Speed measurement schematic based on tachogenerator sensor.

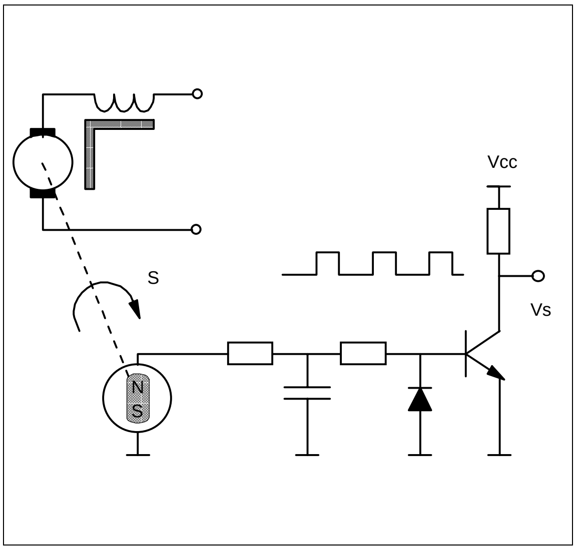
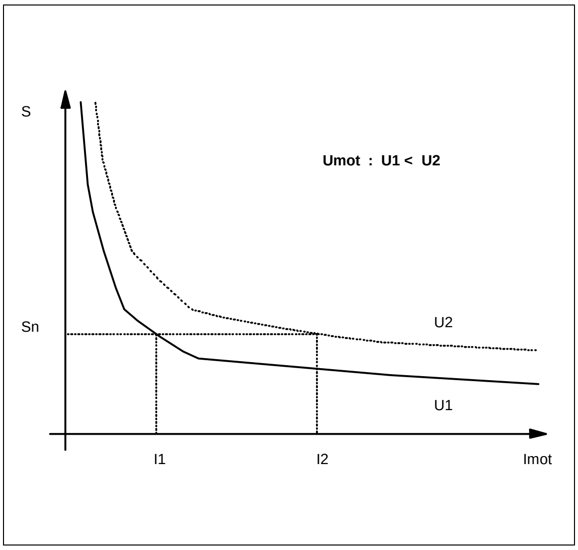


Figure 6. Evaluation of the speed S versus current Imot & voltage Umot of the



The back electromotive force (b.e.m.f.) of the motor is a function of the speed and of the current. In the first approach it behaves as a resistance proportional to speed.

When the sensor is removed, the speed of the motor is determined by relating the average motor voltage and the average motor current. The controller defines the motor voltage by the triac triggering time (a.c. mode), or by the chopper duty cycle (d.c. mode). A shunt resistor allows the peak motor current to be sensed.

Such a control method is feasible despite potential motor saturation and the brush voltage. The relations are not linear, however a microcontroller can solve the relations by using look up tables for calculations.

To improve the behavior of the speed drive on dynamic operations, the controller can also consider the motor acceleration: this acceleration is represented by the motor current variation, ΔI_{MOT}.

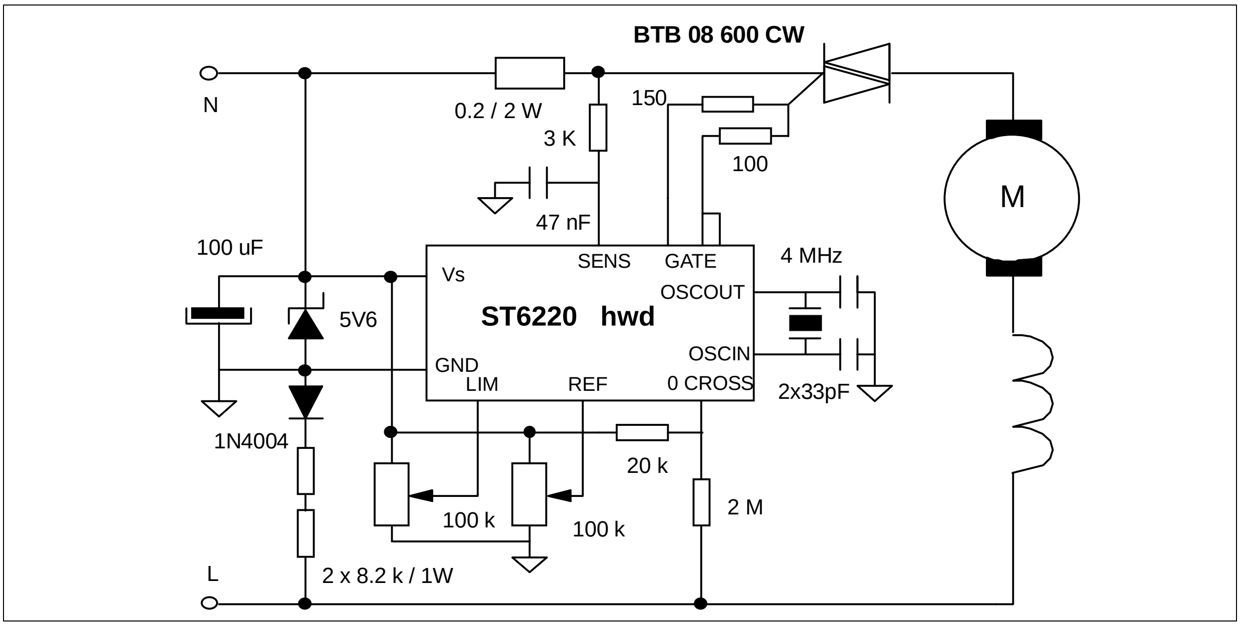
This control method can be applied to the home appliance applications when the requirement on the speed accuracy is not very high. The sensor is not required, so the cost of the drive is reduced and its reliability is improved.

2 AN APPLICATION EXAMPLE

A basic speed drive has been designed with a 500 W a.c. universal motor. An 8 Amp - 600 V SNUBBERLESS triac drives the motor from the 230 V - 50 Hz mains. This drive is adapted to a drill application.

The speed drive control is fulfilled by an ST6220 microcontroller. This 8-bit microcontroller is able to calculate and to control the motor speed with no external sensor by using its on-chip analog/digital converter (A/D) for the current measurement and its 8-bit timer for the triac triggering control.

Figure 7. Application diagram of a speed drive for an 500 W a.c. universal motor.



A shunt resistor measures the peak motor current: it is connected in series with the triac, and is referred to the positive supply polarity. Two potentiometers define the speed reference and the torque limit. Two resistors allow the zero crossing of the line voltage to be captured.

The microcontroller triggers the triac directly with its 20 mA outputs. At the triggering point three outputs supply a 50 mA gate current during 2 instruction cycles (24 µs): this pulse will secure the triggering at low temperature or on accidental dI/dt. The two outputs then remain on during 500 µs, supplying the gate triggering current (I_{GT} = 35 mA) until the triac latches. The triac driver consumption then becomes less than 2 mA.

An auxiliary supply generates a voltage of 5.6V: the low consumption of the HCMOS microcontroller and the pulse mode triac triggering minimize the total consumption (I_{CC} < 5 mA). So the required current is supplied by two 8.2 kΩ/1W decoupling resistors.

Figure 8a. Triac triggering with double pulse mode: diagram.

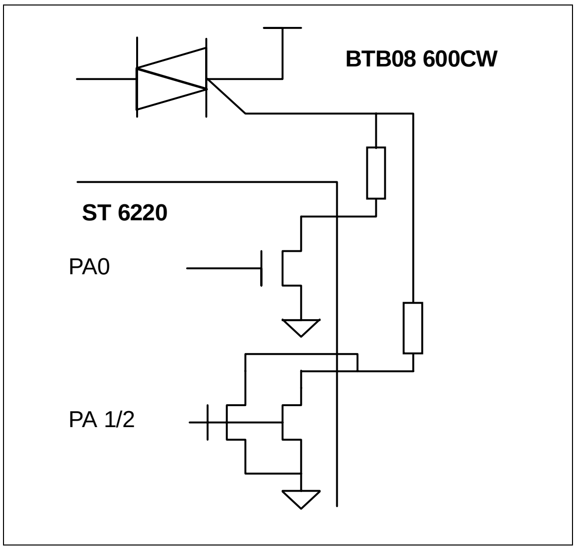
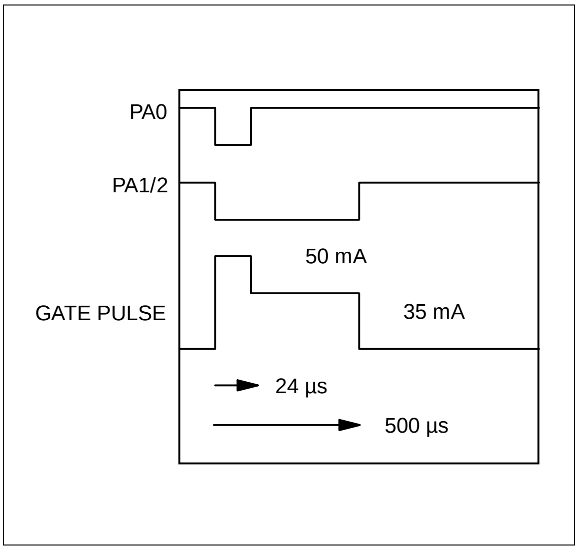
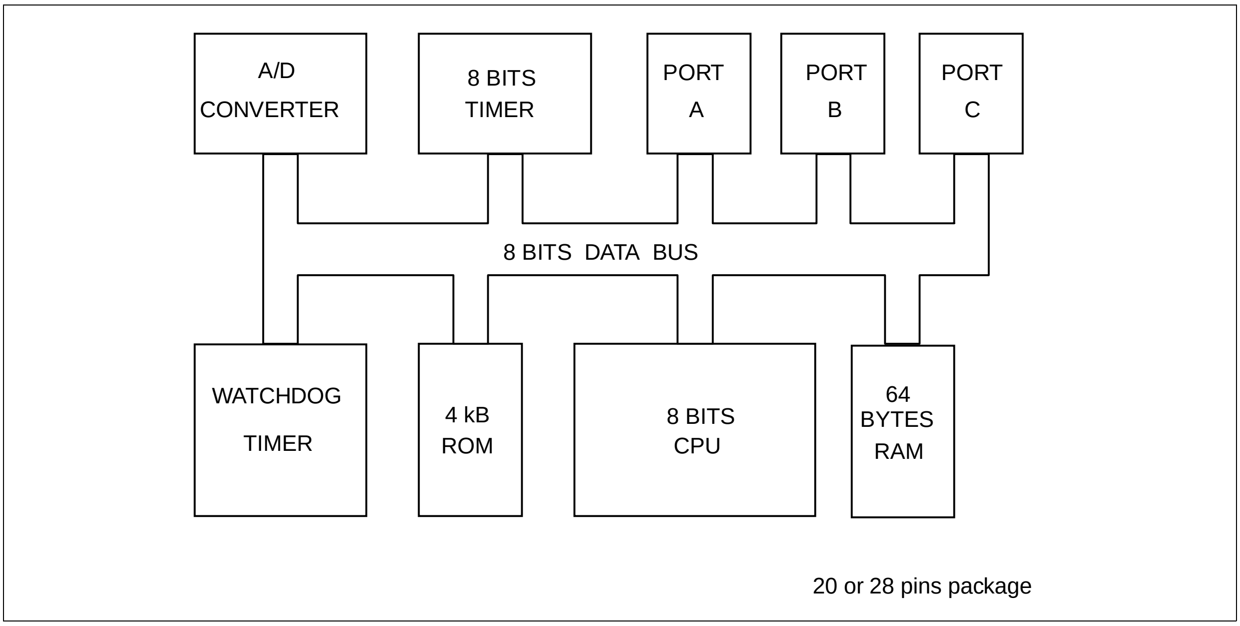


Figure 8b. Triac triggering with double pulse mode: chronogram.



The control program achieves speed control and torque limitation. In addition to these functions, a current measurement task and a.c. phase control are also made by the software.

Figure 9. Block diagram of the ST622x microcontrollers.

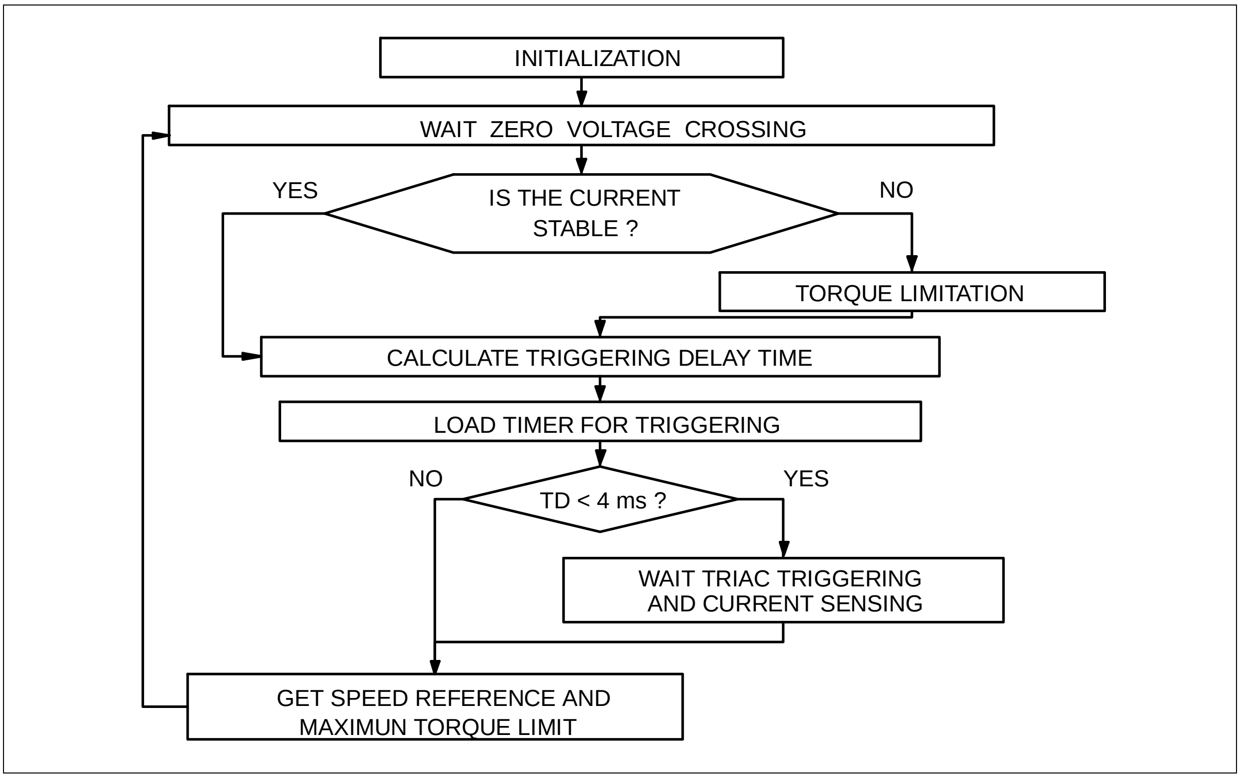


The speed control determines the motor voltage to be set and the triac triggering delay time T_D. At each mains cycle the A/D converter reads the value of the first potentiometer with a 64 step scale. This value defines the speed reference by the means of a 64 byte look up table.
The controller compensates for the effect of the motor current on the speed: it determines the current correction through a 64 byte look up table.

Finally it calculates the time T_D by a combination between the speed reference and the current correction.

The timer organizes the phase angle control. It is synchronized to the zero crossing of the mains voltage. It delays the triac triggering to T_D with an 0.5 % resolution and then generates the 500µs gate triggering pulse.

Figure 10. Flow chart of the speed control algorithm.



The motor current measurement is managed by software, saving the need for external peak detector components. After the triac is triggered, the timer synchronizes the A/D converter to read the shunt resistor. This read is done when the peak motor current is maximum. The peak current instant time T_C is determined versus the previous motor current value by the means of a 64 bytes look up table.

Figure 11. Flow chart of the 8 bits timer subroutine operation.

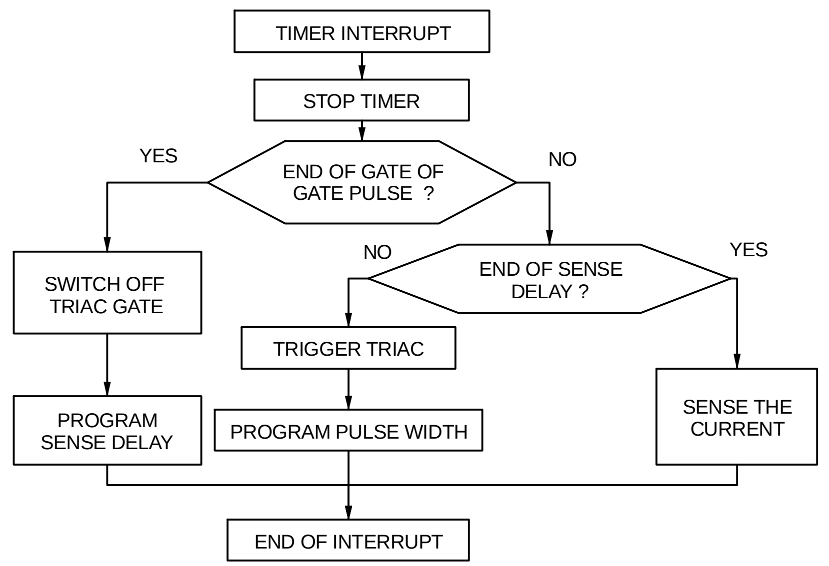
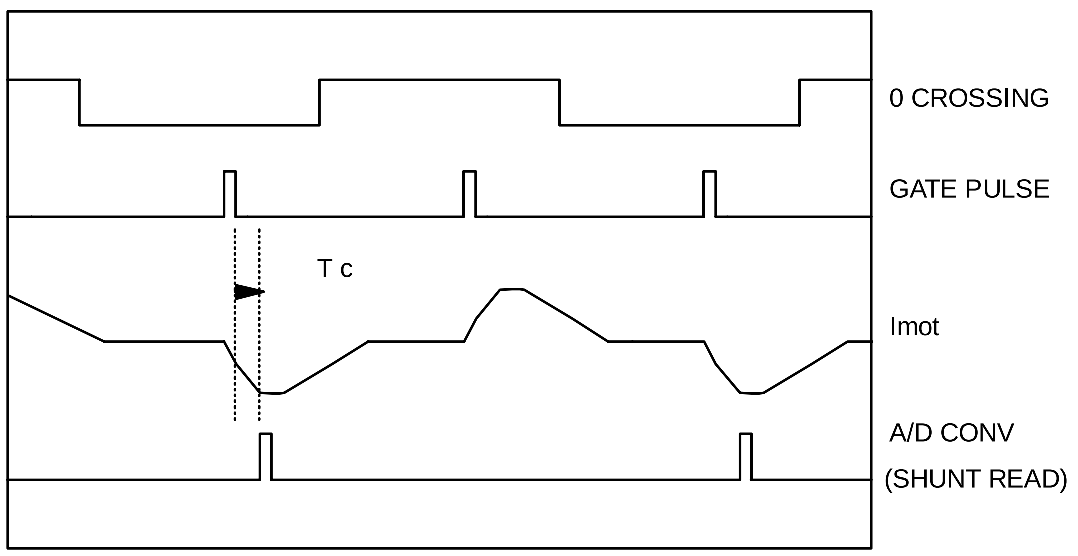


Figure 12. Measurement of the peak motor current with software peak detector.



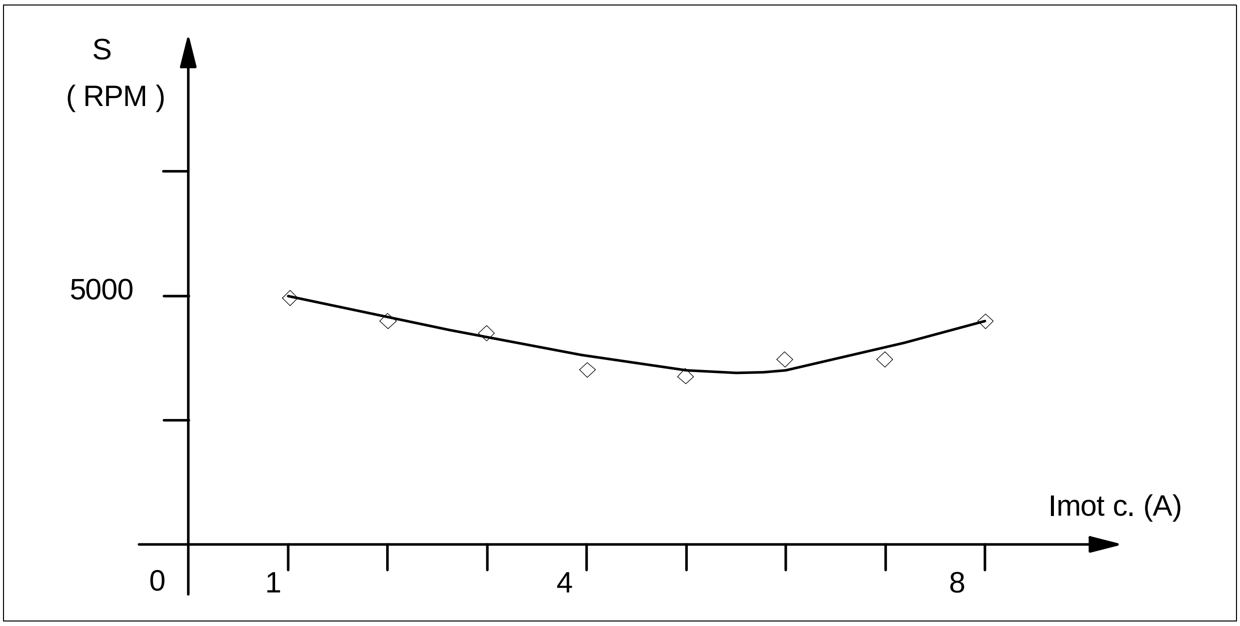
The torque limitation controls the applied force of the drilling tool. The A/D converter reads the value of the second potentiometer to determine the requested torque limit on a 64 step scale. When the motor current is higher than this limit, the motor voltage is limited to a maximum value by limiting the delay time T_D.

The total controller program occupies 640 bytes of ROM Memory, including the look up tables of the speed reference, of the current correction, and of the current sense delay time.

This simple program is designed for one application. To change it for another application or motor, only the two look up tables related to the current (128 bytes) need to be modified. This microcontroller plus triac board can thus drive several motor types, or the performance of the board can be optimized for one or several fixed speeds. This flexibility is possible because of the MCU and of its 4 KByte memory size.

The speed of the designed drive ranges from 4000 to 25000 RPM. When the speed reference is 5000 RPM, the speed decreases down to 3500 RPM at 5A peak; it then increases up to 4500 RPM at 8A peak.

Figure 13. Variation of the speed S versus the peak motor current that represents the motor load.



The torque limitation is mainly effective at low speed: the torque can vary greatly and can decrease the quality of work of the tool. At high speed this limitation becomes useless because the torque and the current are naturally limited by the high impedance of the motor.

3 CONCLUSION

This note presents a sensorless speed controller for an a.c. universal motor, using a SNUBBERLESS triac and an ST6220 microcontroller. The use of such a microcontroller permits the designer to reconsider the design of the brush motor speed drives: it also offers other methods to control the motor and simplifies the drive circuit by reducing the number of components used.

Moreover these ST62 microcontrollers increase the flexibility of the designed circuit. The same hardware circuit can fulfill the control of various motor types by changing only two look-up tables. Other functions such as the user interface (keyboard, display) can be easily added by software to the power control.

The same approach can be extended to motor control in D.C. mode where an IGBT/MOSFET chopper and microcontroller control by P.W.M. are designed.

This study has been made with the collaboration of the company B.F.E. (France), which has developped the program and the demonstration board.

4 REFERENCES

[1] - Power control with triacs and ST6210 MCU
AN 392 - Ph. Rabier and L. Perier (SGS-THOMSON Microelectronics)

[2] - Digital control for brush DC motor
T. Castagnet and J. Nicolai (SGS-THOMSON Microelectronics)
International Appliance Technics, May 93

[3] - Controlling a Brush DC motor with an ST6265
AN414 - J. Nicolai and T. Castagnet (SGS-THOMSON Microelectronics)
PCIM Nuremberg, June 93

[4] - Improved universal motor drives
J.M. Bourgeois, J.M. Charreton, and P. Rault (SGS-THOMSON Microelectronics)
AN422 - Improved Universal Motor Drive with ST62

[5] - Improvement in the triac commutation
AN 439 - P. Rault (SGS-THOMSON Microelectronics)

[6] - Data books of " SCRs and TRIACS" (DBSCRTRI/2)
and " ST62XX MICROCONTROLLERS " (DBST6ST/3)
(SGS-THOMSON Microelectronics)

Appendix 1. Sensorless speed control for the universal motor: customization of the control program.

The software of the motor control is provided in appendix 3 and is named sens01.asm. It can be adapted to an application by adjustment of the three look up tables (speed reference, peak current instant time, and current compensation).
During the adjustments of the speed range and of the peak current detection the current compensation should be inhibited by clearing the current correction register INDEX.

Adjustment of the no load speed range

The potentiometer connected to PB2 (pin 13) defines the speed reference S in conjunction with a 64 byte look up table VITT. This reference corresponds to the motor voltage U_{MOT} and to the triggering delay time T_{D_O} at no load.

The table VITT contains all no load delay times T_{D_O} to define the speed range of the drive. The table values are defined by the full and minimum speed operation:

- the minimum triggering delay time (full speed) is defined by the motor power factor; the triac can only be triggered when its anode current is cancelled;

- the maximum triggering delay time (minimum speed) is chosen to keep sufficient motor magnetization, so as to maintain a relationship between motor torque and current.

The true decimal values of the tables are calculated by dividing the triggering delay time T_{D_O} by the basic counting step of the timer (48 µs).

Figure 1. Variation of the no load speed versus the reference voltage given by the speed potentiometer.

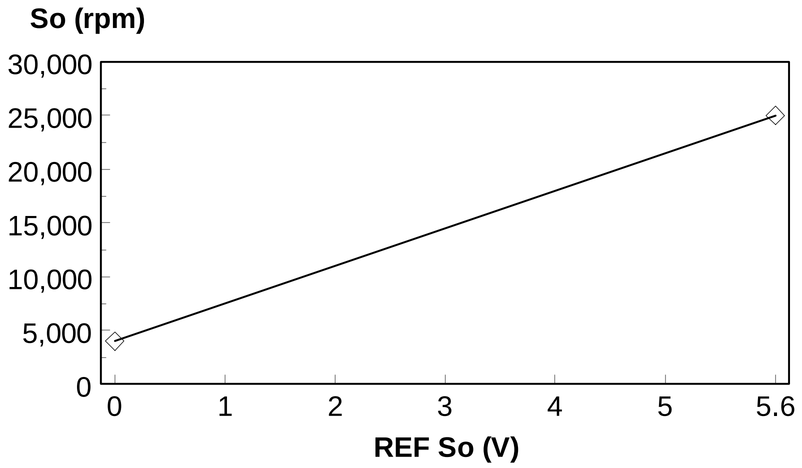
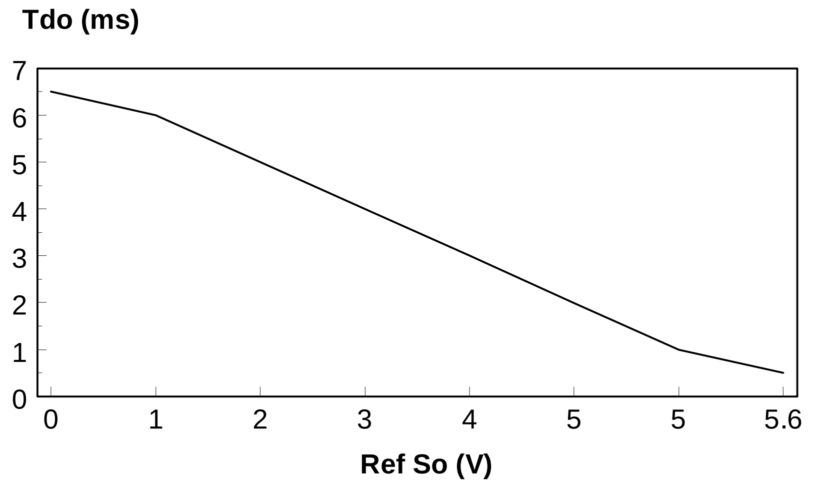


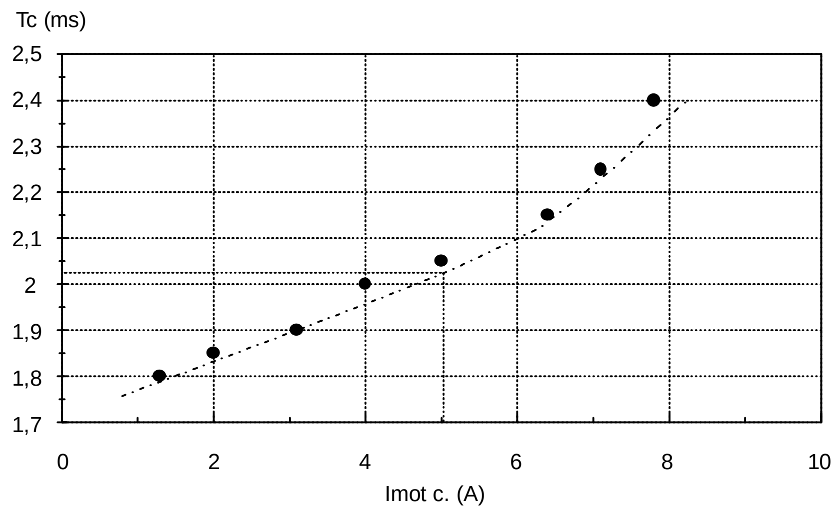
Figure 2. Variation of the no load triac triggering delay time vs the speed reference voltage.



Adjustment of the peak current detection

The peak current detection is made with the A/D converter connected on PB1 (pin 14). The timer synchronizes this operation to the triac triggering. The counted value is issued from a 64 byte look up table RTMES and it is defined versus the previous peak motor current $I_{MOT\ C}$.
The table RTMES is optimized experimentally at the lowest speed S_{MIN}. The peak current instant T_C after triggering is registered by test for several current values which are chosen between 1 A (8d numeric) and 8 Apk (64d numeric). For each case the decimal table value is calculated by subtracting 650µs (triac triggering task duration) to the peak current instant T_C, and by dividing the result by the basic timer step (48 µs).

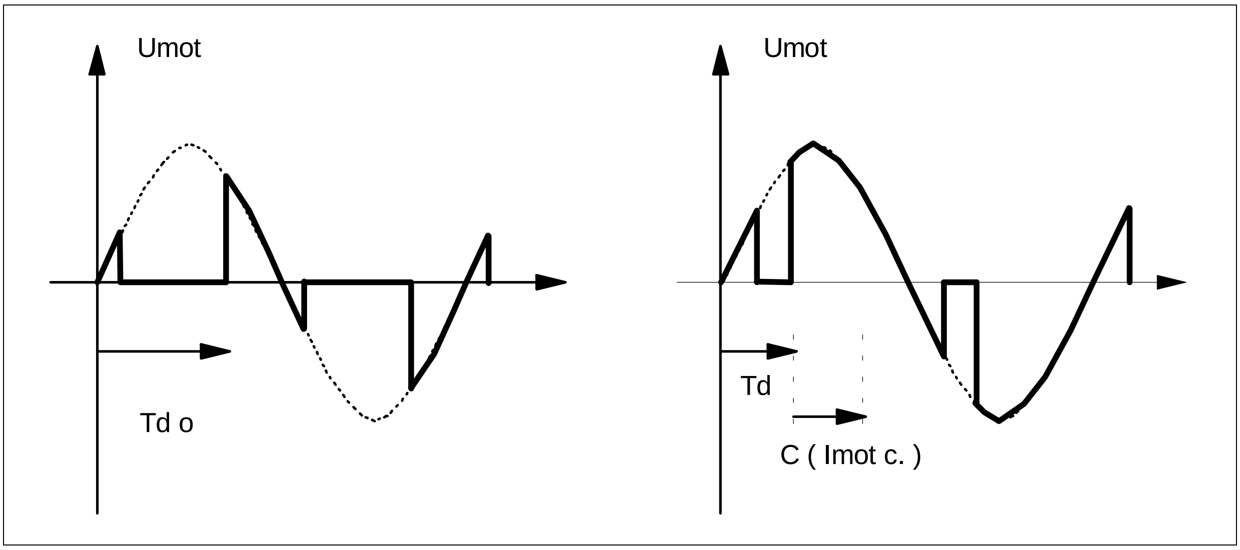
Figure 3. Experimental plotting of the peak current instant T_C versus the peak motor current $I_{MOT\ C}$ at S_{MIN} = 5000 RMP.



The other values of the table are calculated by linear interpolation on these 4 experimental points. The resulting table is fine tuned by a final test. The table is optimized for a speed S_{MIN}, but can extended to a larger speed range.

Adjustment of the current compensation

Figure 4. Motor voltage waveforms with no load and nominal load.

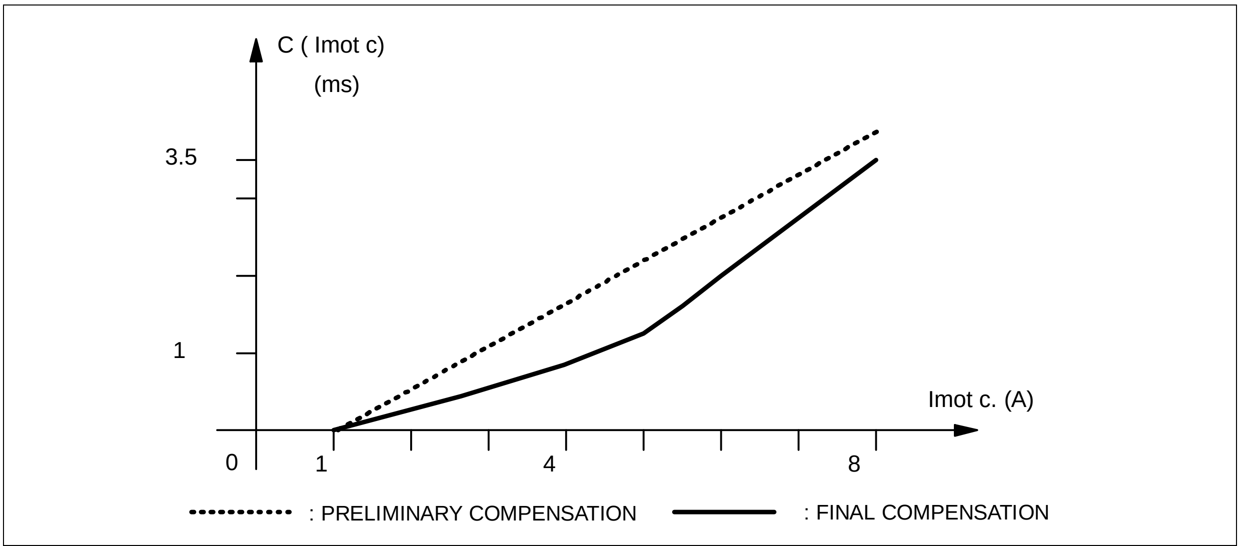


Speed control is done with a basic current compensation. When the load (and the motor current) increases, the controller has to increase the motor voltage: it increases the b.e.m.f. to maintain the motor speed.
The controller defines a current correction versus the peak motor current I_{MOT_C}: $C(I_{MOT\_C})$. The triggering delay time T_D is calculated by subtracting the no load triggering delay time T_{D_O} by this current correction:

$$T_{D\_O} = T_D + C(I_{MOT\_C})$$

The current correction values are stored in the current compensation table COUPLE. This table is optimized for the lowest operational speed S_{MIN}, and its use could be applied to a larger speed range. The table calculation is done in two steps.

Figure 5. Plotting of the current correction C(I_{MOT_C}) issued form the COUPLE look up table.

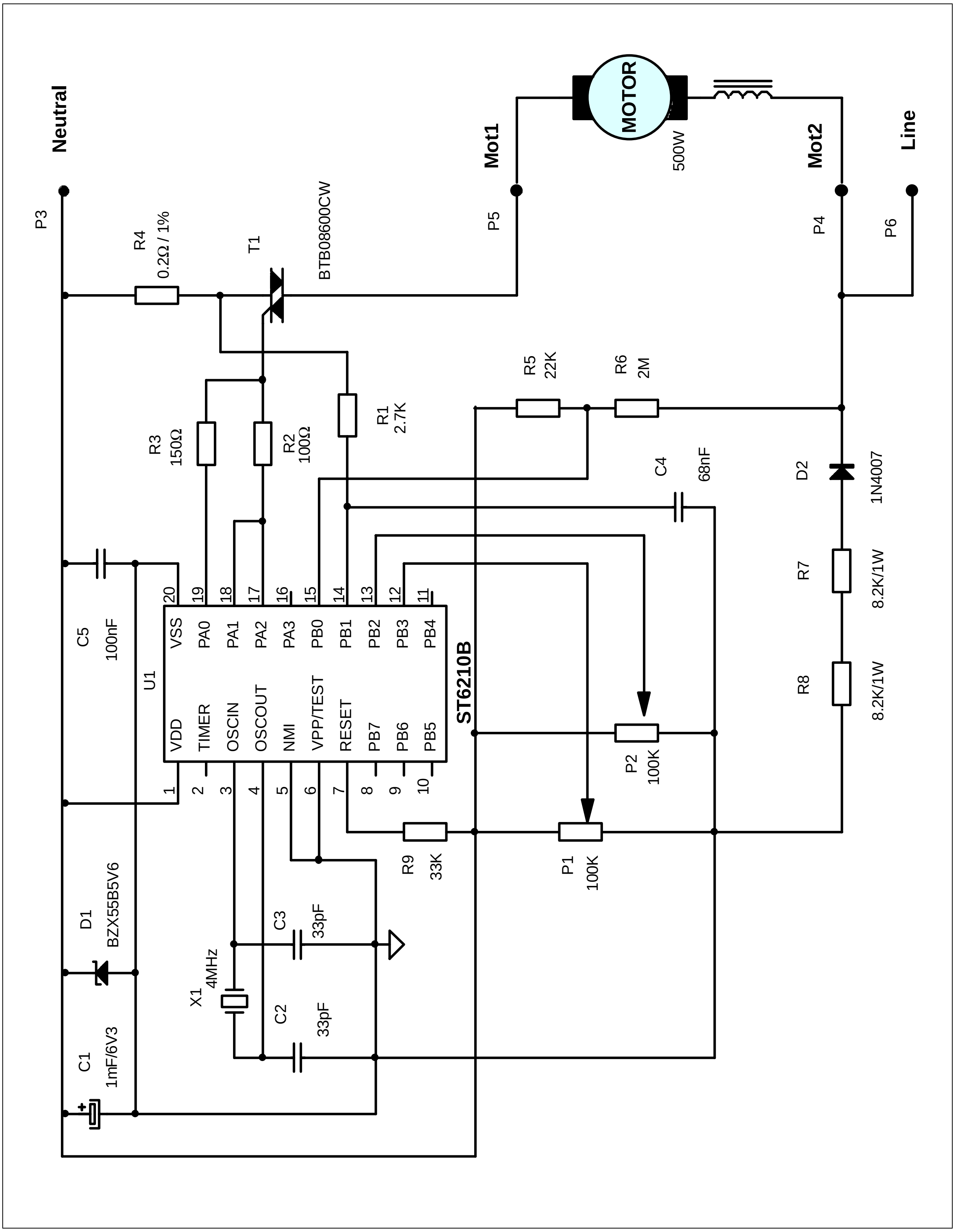


In the first step the current corrections C(I_{MOT_C}) are determined experimentally to obtain a pure compensation of the current influence, and to maintain the speed round S_{MIN}. The INDEX register is loaded with an immediate value C, and the peak motor current is measured on test when the motor speed is at SMIN. The numeric current value defines directly the location of C in the table COUPLE. This test is done for several immediate INDEX values which are chosen between 0 and 80d (3.8 ms). The other values of the table are calculated by linear interpolation on these experimental points.

In the second step the effect of the preliminary defined compensation is reduced on lower motor current operation to give a good speed stability on dynamic operation. The current corrections which are decoded on lower motor current (less than 5 Apk), will be reduced and the resulting table will be tested on the speed drive. On the higher current range the corresponding corrections will be increased to maintain the speed in its operational range.

The resulting table offers a non linear current compensation that gives a good compromise between the speed stability at lower current and the speed decreasing at higher current.

Appendix 2. Circuit diagram



Appendix 3. Software example program.

```
;**************************** SENS 01 ************************************
;*                                                                       *
;*                   SGS THOMSON MICROELECTRONICS                        *
;*                                                                       *
;*************************************************************************
;*              SENSORLESS UNIVERSAL BRUSH MOTOR CONTROL                 *
;*                           VERSION 2.0                                 *
;*                          DECEMBER  1993                               *
;*************************************************************************
;*  This program was developped with the partnership of the company      *
;*  B.F.E. . The address of our consultant is :                          *
;*                          Raymond PORTIER,  B.F.E.                     *
;*        24, avenue du General LECLERC, 65200 Bagneres de Bigorre       *
;*              Tel : (33).62.91.03.00   Fax : (33).62.91.03.87          *
;*************************************************************************
;*  Circuit configuration and key features are :                         *
;*  - ST6220 microcontroller is designed in                              *
;*  - oscillator frequency : 4 MHz                                       *
;*  - hardware watchdog device is implemented                            *
;*  - line zero crossing detection on PB0 with interrupt                 *
;*  - speed reference on PB2 ; torque limitation on PB3                  *
;*  - torque limitation is stopped                                       *
;*  - triac triggering delay time is between 0.4 and 6.5 ms              *
;*  - triac gate drive on PA1, PA2 with boost on PA0                     *
;*  - motor current detection on PB1 with ADC                            *
;*  - current shunt is 0.2 Ohms and detectable peak current is less      *
;*    8 Amps                                                             *
;*  - motor current detection time is shown by PA3 pointer               *
;*  - soft start operation                                               *
;*                                                                       *
;*************************************************************************
.W_ON
;************************* REGISTER DECLARATION ************************

X        .def 080h!M               ; Index register.
Y        .def 081h                 ; Index register.
V        .def 082h                 ; Short direct register.
W        .def 083h                 ; Short direct register.

A        .def 0ffh!M               ; Accumulator.

PRA      .def 0C0h                 ; Port a data register.
PRB      .def 0C1h                 ; Port b data register.

PRAD     .def 0C4h                 ; Port a direction register.
PRBD     .def 0C5h                 ; Port b direction register.•

PRAO     .def 0CCh                 ; Port a option register.
PRBO     .def 0CDh                 ; Port b option register.

IOR      .def 0C8h                 ; Interrupt option register.
DRWR     .def 0C9h!M               ; Data rom window register.

ADR      .def 0D0h!M               ; A/D result register.
ADCR     .def 0D1h                 ; A/D control register.

TPSC     .def 0D2h                 ; Timer 1 prescaler register.
TCR      .def 0D3h                 ; Timer 1 counter register.
TSCR     .def 0D4h                 ; Timer 1 status control register.

WDR      .def 0D8h                 ; Watchdog register.
```

```
;********************** DATA RAM REGISTERS ******************************

VALR    .def 099h!M              ; motor current register
LOOP    .def 087h                ; counter
DX      .def 088h                ; back up of X
DY      .def 089h                ; back up of Y
DXb     .def 08Ah                ; back up of X
DYb     .def 08bh                ; back up of Y
FLIT    .def 08ch                ; motor control flag register
                                 ; b0 indicates 0 crossing pulse
                                 ; b2 indicates timer operation on
                                 ;        triac triggering
                                 ; b3 indicates line polarity versus Vdd
                                 ; b4 indicates timer operation on
                                 ;        current sensing delay
                                 ; b7 indicates timer operation
                                 ; b1, b5, b6 are unused here
DVALR   .def 08eh                ; previous VALR register value
COMPT   .def 08dh                ; soft start counter
DPRB    .def 08fh                ; back up of port B data register
INDEX   .def 091h                ; motor current compensation register
ADCcou  .def 092h                ; torque limitation register
ADCcou1 .def 093h
ADCvit  .def 094h!M              ; speed reference register
ADCvit1 .def 095h
DA      .def 096h                ; back up of A
DAb     .def 097h                ; back up of A

;********************** EQUATE DEFINITION *******************************

OFSET   .equ 008h                ; offset subtracted from motor current
                                 ;    in current compensation calculation
OFSET1  .equ 007h                ; offset subtracted from motor current
                                 ;     in measure delay time calculation
TGATE   .equ 008h                ; triac gate pulse duration  08h=385us
TDTIM   .equ 053h                ; time limit to define priority between
                                 ;     timer & potentiometers subroutines
TDMIN   .equ 008h                ; minimum trig. delay time 08h=385us
TDMAX   .equ 09ch                ; maximum trig. delay time 9ch=9.0 ms
START   .equ 002h                ; step of soft start operation

;****************** BEGINNING OF PROGRAM AREA ***************************
        .org 0800h
;***********************************************************************
;*                   SPEED REFERENCE TABLE                             *
;***********************************************************************

VITT    .BYTE   86H,84h,82H,80H,7eH,7cH,7aH,78H
        .BYTE   76H,74h,72H,70H,6eH,6cH,6aH,68H
        .BYTE   66H,64h,62H,60H,5eH,5cH,5aH,59H
        .BYTE   58h,57H,56H,55H,54H,53H,52H,51H
        .BYTE   4fH,4dH,4bH,49H,47H,45H,43H,41H
        .BYTE   3fH,3dH,3bH,39H,37H,35H,33H,31H
        .BYTE   2fH,2dH,2bH,29H,27H,25H,23H,21H
        .BYTE   1fH,1dH,1bH,19H,17H,15H,13H,11H

;***********************************************************************
;*                   CURRENT COMPENSATION TABLE                        *
;***********************************************************************

COUPLE  .BYTE   00H,01H,01h,02H,03H,04H,04H,05H
        .BYTE   06H,07H,07H,08H,09H,0ah,0aH,0bH
        .BYTE   0cH,0dH,0dh,0eH,0fH,10H,10H,11H
        .BYTE   12H,13H,13H,14H,15H,16H,16H,17H
```

```
        .BYTE   18H,19H,19h,1aH,1bH,1cH,1dH,1eH
        .BYTE   1fH,20H,21H,22H,23H,24H,25H,26H
        .BYTE   27H,29H,2ah,2bH,2cH,2eH,2fH,30H
        .BYTE   31H,33h,34H,35H,37H,38H,3bH,3eH

;***************************************************************************
;*                PEAK CURRENT INSTANT TIME TABLE                          *
;***************************************************************************

RTMES   .BYTE   0aH,0ah,10H,10H,10H,11H,11H,11H
        .BYTE   12H,12H,12H,13H,13H,13H,14H,14H
        .BYTE   14H,14H,14H,15H,15H,15H,15H,15H
        .BYTE   16H,16H,16H,16H,16H,17H,17H,17H•
        .BYTE   17H,17H,17H,18H,18H,18H,18H,18H
        .BYTE   18H,19H,19H,19H,19h,19H,1aH,1aH
        .BYTE   1aH,1aH,1bH,1bh,1bH,1cH,1cH,1cH
        .BYTE   1dH,1dH,1eH,1eH,1fH,20H,20H,21H

;***************************************************************************
;*                          INITIALIZATION                                 *
;***************************************************************************

start   ldi     WDR,0feh           ; watchdog initialization

        ldi     X,084h
raz     clr     A                  ; clear the RAM
        ld      (X),A
        inc     X
        ld      A,X
        cpi     A,0d5h
        jrc     raz

INIT    ldi     PRA,   0fh         ; port A in push pull output
        ldi     PRAD,  0fh         ; connected at Vdd
        ldi     PRAO,  0fh

        ldi     PRB,   0eh         ; PB0 in interrupt input, PB1 in analog
        ldi     PRBD,  00h         ; input, PB4/5/6/7 in pull up inputs,
        ldi     PRBO,  03h         ; PB2/3 in HI input

        ldi     ADR,00h            ; A/D conv. initialization
        ldi     ADCR,00h           ; ADC is stopped

        ldi     FLIT,   00h        ; clear logic and application registers
        ldi     COMPT,  0ah        ;
        ldi     INDEX,  00h        ;
        ldi     VALR,   00h        ;
;       ldi     DVALR,  00h        ;
        ldi     ADCvit, 09fh       ;

        ldi     IOR,10h            ;  interrupt validation
                                   ;  PB0 interrupt on falling edge
        reti
;***************************************************************************
;*                          MAIN PROGRAM                                   *
;***************************************************************************

;********************** SOFT START TASK ******************************

SOFT1   jrr     0,FLIT,SOFT1       ; wait 0 crossing ( # 1 )
        res     0,FLIT
        call    VIT
        ldi     A,TDMAX
;       ld      A,ADCvit           ; ENABLE THIS INSTRUCTION TO INHIBIT
```

```
                                  ; THE SOFT START
SOFT2   jrr     0,FLIT,SOFT2      ; wait 0 crossing ( # 2,.., n-1 )
        res     0,FLIT            ; reset b0 of FLIT
        ld      TCR,A             ; load timer register
        ldi     TSCR,01111100b    ; start timer with interrupt & PSC = 16
        set     7,FLIT            ; control timer cycle with b7 indicator
ATINI   jrs     7,FLIT,ATINI      ; wait end of timer operation
        subi    A, START
        cp      A, ADCvit
        jrnc    SOFT3
        jp      SOFT4
SOFT3   jp      SOFT2
SOFT4   ld      A, ADCvit

;***************** MAIN MOTOR CONTROL PROGRAM ****************************

MAIN    jrr     0,FLIT,MAIN       ; wait 0 crossing ( # n )
        res     0,FLIT

MAIN1   ldi     DRWR,COUPLE.W     ; define motor current compensation
        ld      A,   VALR         ; A <= VALR motor current measure
        subi    A,   OFSET        ; A <= VALR - OFSET     ,
        jrnc    MAIN4
        clr     A

MAIN4   cpi     A,  040h          ; check VALR to max. value
        jrc     MAIN3
        ldi     A,  03fh          ; limit VALR to its max. value

MAIN3   addi    A,  040h
        ld      X,   A            ; load VALR@ in X register
MAIN2   ld      A,   (X)          ; calculate current compensation data

;            ****** TORQUE LIMITATION TASK ******

        jp      MAINc             ; ENABLE THIS INSTRUCTION TO STOP
                                  ;     TORQUE LIMITATION
        cp      A,   ADCcou       ;
        jrc     MAINc
        ld      A,   ADCcou       ; limit to the max. requested torque

;            ******** SPEED CONTROL TASK ********

MAINc   ld     INDEX,A

;       ldi    INDEX,00h          ; ENABLE THIS INSTRUCTION TO INHIBIT
                                  ; CURRENT COMPENSATION

MAIN5a  ld      A,    ADCvit      ; load speed reference
        sub     A,    INDEX       ; substrate current comp. to speed ref.
        jrnc    MAIN5
        ldi     A,    TDMIN       ; limit trig. delay time to min. value

;               ***** PHASE ANGLE CONTROL TASK *****

MAIN5   ld      TCR,A             ; load timer for triac triggering delay
        ldi     TSCR,01111100b    ; start timer with interrupt & PSC = 16
        set     7,FLIT            ; b7 <= 1, b7 indicates timer operation

;                 ****** DELAY TIME CONTROL ******

        cpi     A,   TDTIM        ; if Tdelay > TDMIN,
        jrc     MAINm             ; then read references before triggering
```

```
;         ****** SPEED REFERENCE & TORQUE LIMIT TASK *****

        jrr     3,FLIT,ATfin        ; read potentiometer when Vac is > 0
        call    VIT                 ; read speed ref. & torque lim.
ATfin   jrs     7,FLIT,ATfin        ; wait end of timer operation
        JP      MAIN

MAINm   jrs     7,FLIT,MAINm        ; wait end of timer operation
        jrr     3,FLIT,FIN          ; read potentiometer when Vac is > 0
        call    VIT                 ; read speed ref. & torque lim.
FIN     jp      MAIN

;***************************************************************************
;*                      PROGRAM SUBROUTINES                                *
;***************************************************************************

;*************** CURRENT MEASUREMENT SUBROUTINE ************************

ADC     ldi     PRB,   0eh          ; PB1 in A/D input
        ldi     PRBD,  00h
        ldi     PRBO,  03h
        ldi     ADCR,  30h          ; start conversion
ADC1    jrr     6,ADCR,ADC1         ; wait end of conversion
        ld      A,     ADR          ; A <= ADR
        com     A                   ; complement A/D result to obtain
                                    ; current measure referred to Vss
ADC2    ld      VALR,    A          ; update motor current measure in VALR
        ret

;******** SPEED REFERENCE & TORQUE LIMITATION MEASURE SUBROUTINE *********

;            ****** SPEED REFERENCE MEASUREMENT TASK *******

VIT     ldi     PRB,   0eh          ; PB2 input connected on A/D converter
        ldi     PRBD,  00h
        ldi     PRBO,  05h
        ldi     ADCR,  30h          ; A/D conversion start
VITadc  jrr     6,ADCR,VITadc       ; wait at end of conversion
        ld      A,      ADR
        ld      ADCvit, A
        ldi     X,      ADCvit
        ldi     ADCvit1,00h
        call    DIV4

        ldi     DRWR,VITT.W         ; convert measured value in
        ld      A,     ADCvit1      ; triac triggering delay time
        addi    A,     40h
        ld      X,     A
        ld      A,    (X)
        ld      ADCvit,A

;          ****** TORQUE REFERENCE MEASUREMENT TASK ******

COU     ldi     PRB,    0eh         ; PB3 input connected on A/D converter
        ldi     PRBD,   00h
        ldi     PRBO,   09h

        ldi     ADCR,   30h         ; A/D conversion start
COU1    jrr     6,ADCR, COU1        ; wait at end of conversion
        ld      A,      ADR
        ld      ADCcou, A
        ldi     X,      ADCcou
        ldi     ADCcou1,00h
        call    DIV4
```

```
        ldi     DRWR,COUPLE.W      ; convert measured value in
        ld      A,      ADCcou1    ; triac triggering delay time
        addi    A,      40h
        ld      X,      A
        ld      A,    (X)
        ld      ADCcou,A

FVIT    ldi     PRB,    0eh
        ldi     PRBO,   03h
        ret
;***************** DIVISION BY 4 SUBROUTINE *****************************

DIV4    ldi     LOOP,   06h
DIV42   ld      A,(x)
        sla     A
        ld      (X),A
        inc     X
        ld      A,(X)
        rlc     A
        ld      (X),A
        dec     LOOP
        jrz     DIV41
        dec     X
        jp      DIV42
DIV41   ret

;************* REGISTER CONTEXT SAVING SUBROUTINE ************************

SR      ld      DA, A              ; A-->DA
        ld      A,  X              ; X-->A
        ld      DX, A              ; A-->DX
        ld      A,  Y              ; Y-->A
        ld      DY, A              ; A-->DY
        ret

;************* REGISTER CONTEXT RESTORING SUBROUTINE *********************

RSTR    ld      A,DX               ; DX-->A
        ld      X,A                ; A-->X
        ld      A,DY               ; DY-->A
        ld      Y,A                ; A-->Y
        ld      A,DA               ; DA-->A
        ret

;************* REGISTER CONTEXT SAVING SUBROUTINE ************************

SRb0    ld      DAb,A              ; A-->DAb
        ld      A,  X              ; X-->A
        ld      DXb,A              ; A-->DX
        ld      A,  Y              ; Y-->A
        ld      DYb,A              ; A-->DY
        ret

;************* REGISTER CONTEXT RESTORING SUBROUTINE *********************

RSTRb0  ld      A,DXb              ; DX-->A
        ld      X,A                ; A-->X
        ld      A,DYb              ; DY-->A
        ld      Y,A                ; A-->Y
        ld      A,DAb              ; DAb-->A
        ret

;***************** TIMER INTERRUPT SUBROUTINE ***************************
```

```
ITIM    ldi     TSCR,00h            ; stop the timer
        call    SR                  ; save context

        jrs     2,FLIT,ITIM1        ; 2nd interrupt ?
        jrs     4,FLIT,ITIM2        ; 3rd interrupt ?

        set     2,    FLIT          ; 1st interrupt  ;  b2 <= 1
        ldi     PRA,  00h           ; TRIGGER THE TRIAC WITH BOOST
        nop                         ;
        ldi     PRA, 01h            ; REDUCE GATE CURRENT and wait at triac
                                    ; latching ( PA0 turns off )
        ldi     TCR,  TGATE         ; load timer triac triggering
        ldi     TSCR, 01111100b     ; start timer & PSC = 16
        jp      FTIM

ITIM1   res     2,    FLIT          ; 2nd interrupt
        ldi     PRA,  07h           ; STOP THE TRIAC GATE PULSE : pA to Vdd
        set     4,    FLIT          ; prepare 3rd interrupt

        ldi     DRWR, RTMES.W       ; current measure delay time
        ld      A,    VALR
        subi    A,    OFSET1        ; A <= VALR - OFSET1
        jrnc    ITIM3
        clr     A
ITIM3   cpi     A,    040h
        jrc     ITIM4
        ldi     A,    03fh
ITIM4   addi    A,    040h
        ld      X,    A
        ld      A,    (X)

        ld      TCR,  A             ; load timer with maesure. delay time
        ldi     TSCR,01111100b      ; start timer & PSC = 16
        jp      FTIM

ITIM2   res     4,    FLIT          ; 3rd interrupt
        jrs     3,FLIT,FTIM1        ; sense motor current when Vac < 0

        ldi     PRA, 0fh            ; pointer for current measure test
        call    ADC                 ; if negative, measure shunt voltage
        ldi     PRA, 07h            ; end of pointer ( optional )

FTIM1   res     7,   FLIT           ; END OF TIMER OPERATION

FTIM    call    RSTR                ; restore context
        reti
;****************** 0 CROSSING INTERRUPT SUBROUTINE **********************

IPB     ldi     WDR,   0feh         ; watchdog control
        call    SRb0                ; save context
        jrr     3,FLIT,IPB1         ; test on line half cycle polarity

        res     3,     FLIT         ; negative half cycle operation
        ldi     IOR,   30h          ; prepare rising edge interrupt
        jp      IPB2

IPB1    set     3,     FLIT         ; positive half cycle operation
        ldi     IOR,   10h          ; prepare falling edge interrupt

IPB2    set     0,     FLIT         ; 0 crossing indicator validation
        call    RSTRb0              ; restore context
        reti

;******************** UNUSED INTERRUPT ADDRESSES ***********************
```

```
IADC    reti
IPA     reti
IMNI    reti

;*********************** INTERRUPT VECTORS *******************************
        .org 0ff0h

        jp      IADC            ;  adc
        jp      ITIM            ;  timer
        jp      IPB             ;  port b and c
        jp      IPA             ;  port a

        .org 0ffch

        jp      IMNI            ;  non maskable interrupt vector
        jp      start           ;  reset interrupt vector
;***************************************************************************

        .eject
        .end
```

THE SOFTWARE INCLUDED IN THIS NOTE IS FOR GUIDANCE ONLY. SGS-THOMSON SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM USE OF THE SOFTWARE.