AN1607
APPLICATION NOTE

PHASE ANGLE PARTIALIZATION MOTOR CONTROL WITH ST52x440

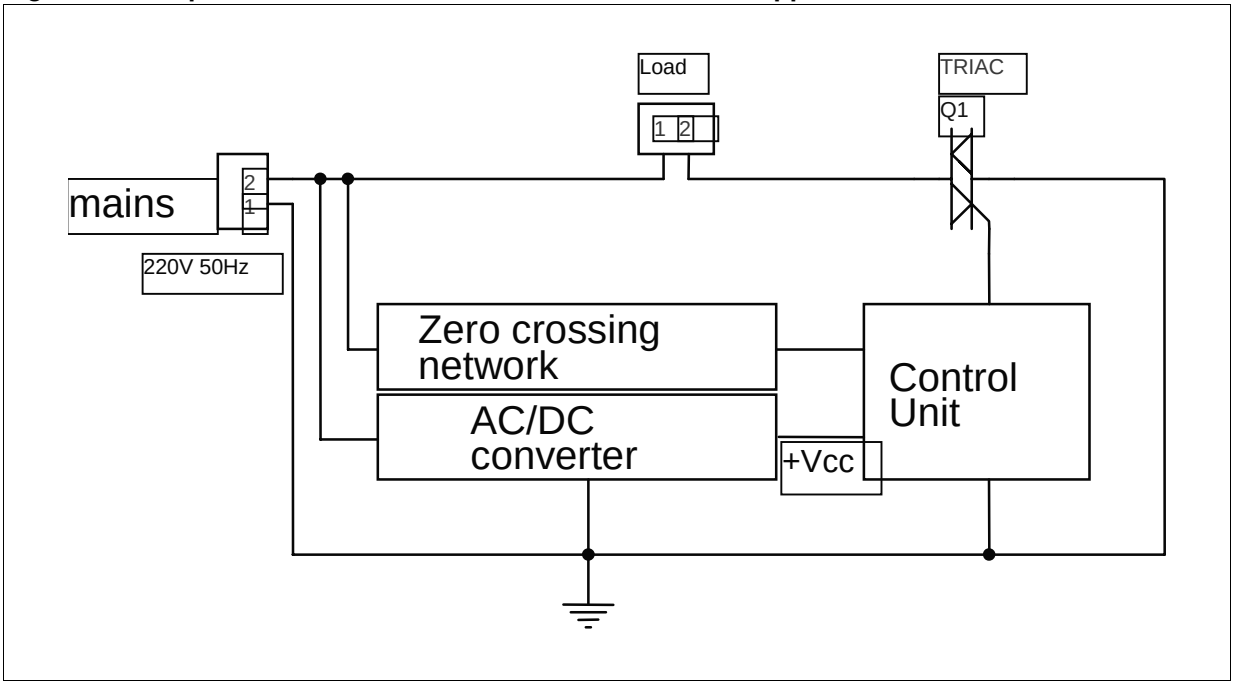
Author: R.Gambino, G. Di Marco

1 INTRODUCTION

This application note describes a simple method to drive a Triac in order to implement a phase angle partialization control. In particular, we want to point out how easily this kind of control can be achieved with ST52x440.

The principle circuit schematic described in this application note is shown in figure 1:

Figure 1 Principle schematic of the circuit described in this application note



1 AC/DC converter is the power supply for the control unit. This power supply has been created by using a Viper12 device;

2. Zero Crossing Network provides the Control Unit with a TTL signal in according to the zero crossing of the mains and ST52x440 allows to use only one resistor. Triac is working on the second and third quadrants of its characteristics (i.e. negative current pulses on the gate switch on the device);

3.Control Unit consists of an ST52x440 device and external circuitry.

TABLE OF CONTENTS

1 INTRODUCTION 1
2 HOW THE PHASE ANGLE PARTIALIZATION WORKS 3
3 ST52x440: A SUITED DEVICE FOR TRIAC DRIVING. 4
4 THE HARDWARE 5
5 THE SOFTWARE 6
5.1 Hardware configuration of pins and peripherals. 7
5.2 Main program. 10
5.3 ADC Interrupt Service Routine. 10
6 EXPERIMENTAL RESULTS 13
6.1 Zero Crossing Detection. 13
6.2 Current on the load. 14
6.3 All signals 16
7 ASSEMBLER CODE 17
APPENDIX A. 23
References:..... 24
BOM..... 25

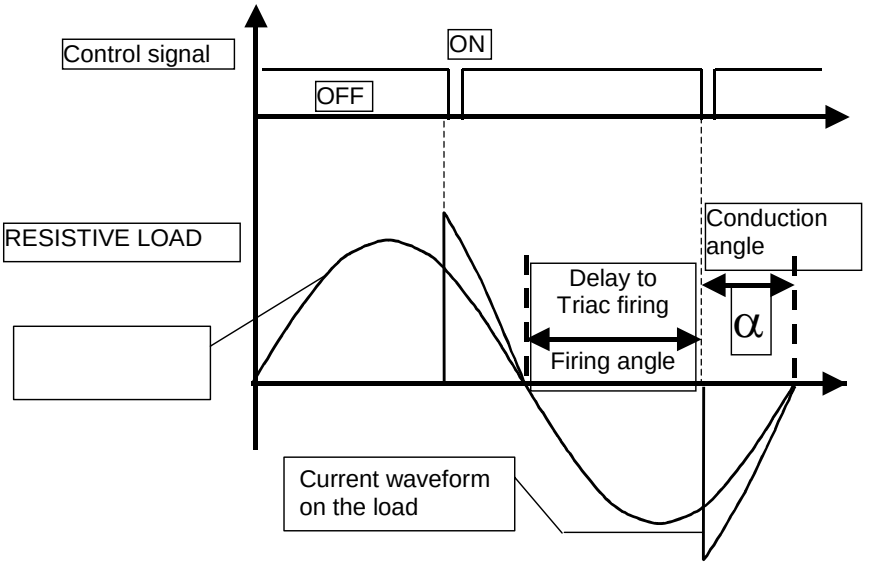
2 HOW THE PHASE ANGLE PARTIALIZATION WORKS

This kind of control, the current is supplied from the mains to the resistive or inductive load for a part of each positive and negative half sinusoyd of mains voltage; thus the RMS voltage applied to the load and hence the power changes (fig.2).

In details, the Triac is switched on after the mains voltage zero crossing with a delay between 0 and 10 msec at the mains frequency of 50Hz.

In this way, a current pulse is supplied to the load; it starts when the Triac switches on and ends at the Triac current zero crossing. The current pulse depends on the temporal distance between mains voltage zero crossing and the Triac firing.

Figure 2 : Phase Angle Partialization.



At zero crossing, the peripheral waits a time interval corresponding to firing angle and then it switches on the Triac. This example refers to resistive loads (no delay between current and voltage waveforms on the load).

Obviously the power on the load changes with this temporal distance: the higher is the delay from zero crossing to Triac firing and the lower is the power provided to the load and vice versa.

This delay can be measured with an equivalent angle that varies from 0° to 180° degrees corresponding to the time delay from 0 to 10 msec.

From this point on, this angle will be called **firing angle**, though in literature also **conduction angle** can be found (angle a of fig. 2): it is the complementary angle of the firing one. The power supplied to the load is strictly related to the firing angle.

If the load is a single phase AC motor, to avoid the presence of a continuous voltage component in the motor, the firing angle must be the same for positive and negative mains waveforms (symmetric control). It means that the firing angle can be changed only once in a period of the mains (every 20msec if 50Hz). Figure 2 shows the control signal used to fire the Triac: when the Triac is working in the second and third quadrant of its characteristics, this signal is active low.

3 ST52X440: A SUITED DEVICE FOR TRIAC DRIVING

The phase angle partialitation can be easily achieved by the ST52x440. In fact, this micro is suitable for Triac motor control thanks to the presence of the Triac Driver internal peripheral.

Some interesting features of this peripheral are indicated below:

1. It can work in PWM mode, Burst mode and Phase Angle Partialization mode. In this application note we will refer, in particular, to Phase Angle Partialization working mode.

2. It provides Main1 input.

In order to work well, the exact zero crossing of the main voltage signal should be detected. By using the ST52x440 only a resistor (220kW 1/2W) is enough to achieve it. In this manner the external components on the board are drastically reduced.

3. After each zero crossing detected on MAIN1 pin, the peripheral (if enabled) furnishes eight pulses for the Triac firing on its TROUT pin. The firing angle is automatically calculated by the peripheral based on the value of the TRIAC_COUNT 8-bit register. The lower is the value stored in TRIAC_COUNT, the lower is the firing angle and so the higher the power delivered to the load. On the contrary, the higher is the value stored in TRIAC_COUNT register, the higher is the firing angle and so the lower the power delivered to the load.

Therefore, when the user wants to change the firing angle, he has only to change the value stored in the TRIAC_COUNT, saving timers and resources of the microcontroller.

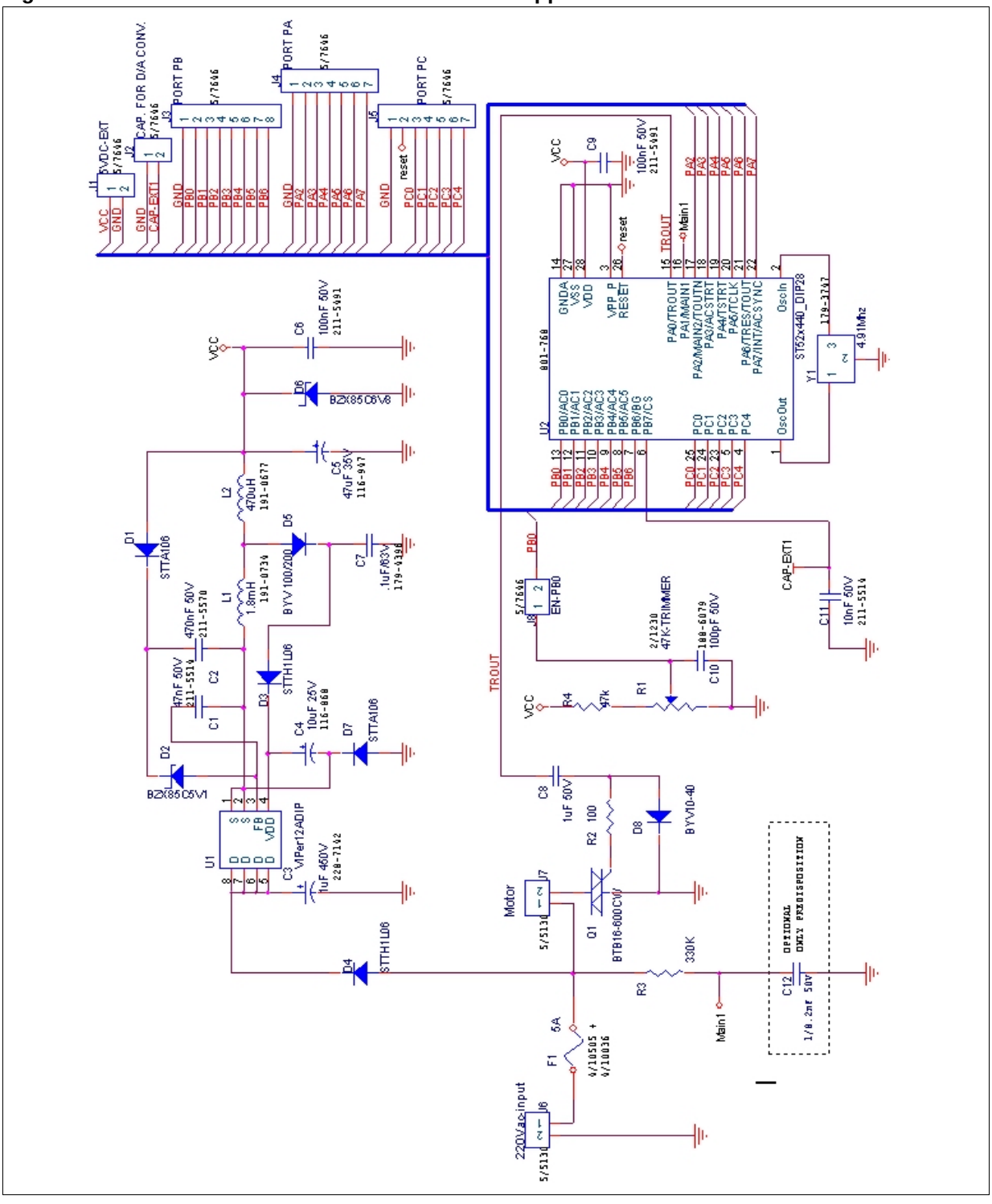
It is very important to notice that the calculus of the firing angle and the generation of the firing pulses for the Triac are executed automatically by the peripheral without using CORE resources.

Moreover, it is important to notice that the TROUT pin can supply up to 50mA; so it can directly drive the Triac gate.

4 THE HARDWARE

Figure 3 shows the schematic of the circuit described in this application note:

Figure 3 :Schematic of the circuit described in this application note



How the board works:

1. A potentiometer is used in order to vary the firing angle. The voltage level is converted by ST52x440 ADC, through PB0 analog input pin, to update the TRIAC_COUNT register.
2. All I/O pins are available on the board, and can be used for general purpose aims.
3. Viper12 with few external components (D4, C3, L1, L2, C5 and D7) works as a Step-Down AC/DC converter to obtain 5V, without voltage regulator (like a 7805).
4. The Triac is switched on by negative pulses applied to its gate. As the Triac A1 pin is grounded, a derivative circuit is mandatory in order to generate the firing pulses. C8 and R2 are this derivative circuit. Its function is to provide a negative pulse for the Triac gate on each falling edge of the TROUT pin. The positive pulses provided by the derivative circuit on each rising edge of the TROUT pin are clamped by diode D8.
5. Every Triac or ACS can be used in this board. The user has only to choose the correct device based on the current to provide to the load. Of course, in this case the fuse has also to be changed. Moreover, the thermal behavior of the Triac and/or ACS must be checked. A heat sink could be necessary for the Triac/ACS.
6. Zero Crossing detection network is constituted of Resistor R3 and the internal protection diodes of MAIN1 pin.

5 THE SOFTWARE

The firmware developed to implement the Triac motor control described on this application note can be downloaded free from ST MCU support site application note section at the following URL's:

http://www.stmcu.com/mcdfiles/AN1607.zip

In chapter 8 you can find the relative assembly file.

This firmware has been developed by using Visual Five 5.0x, the powerful software development tool that can be downloaded, free of charge, from the ST MCU support site at the following URL:

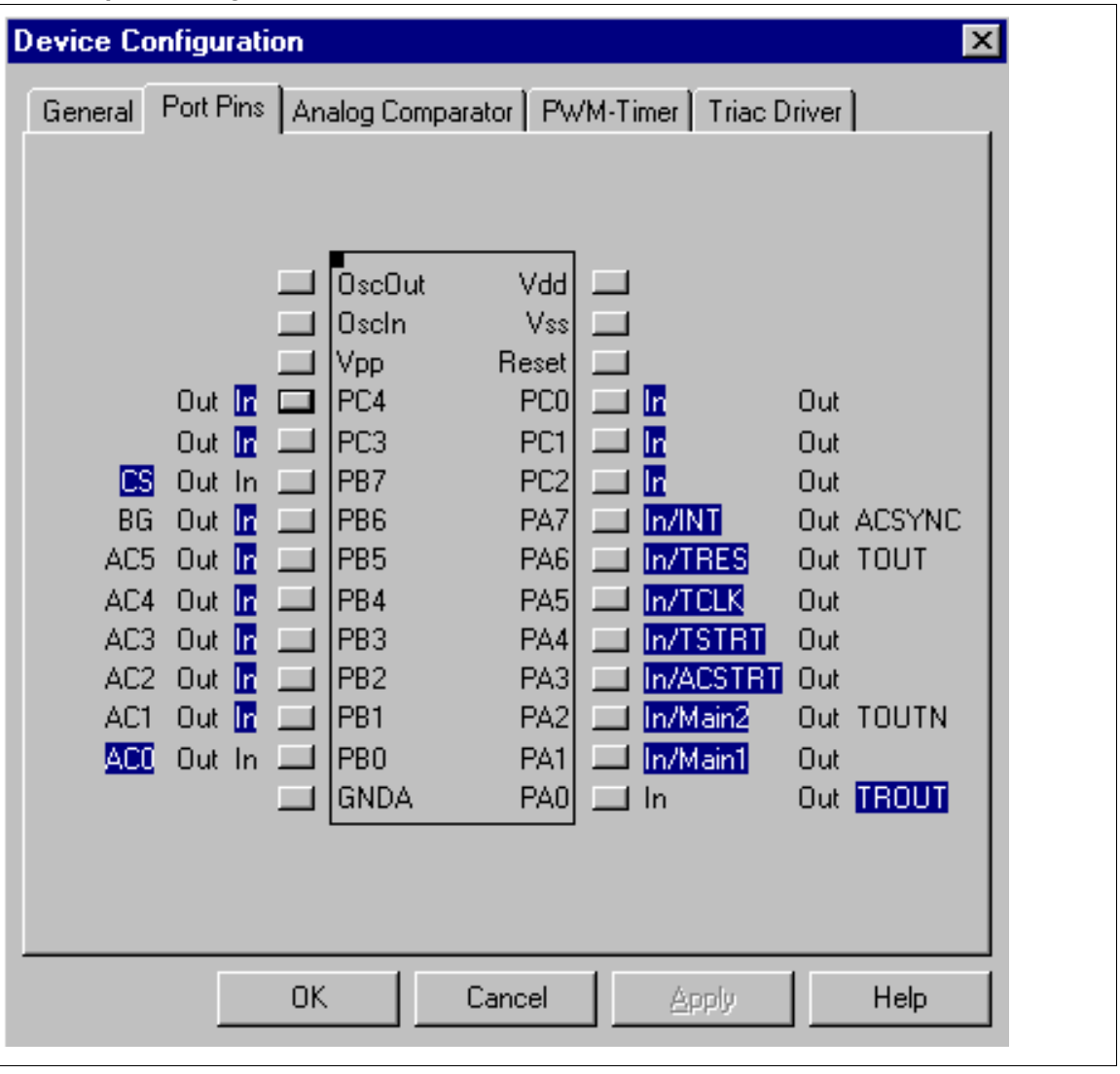
http://www.stmcu.com/familiesdocs-76.html

Development of firmware can be done through the following steps:

5.1 Hardware Configuration of Pins and Peripherals

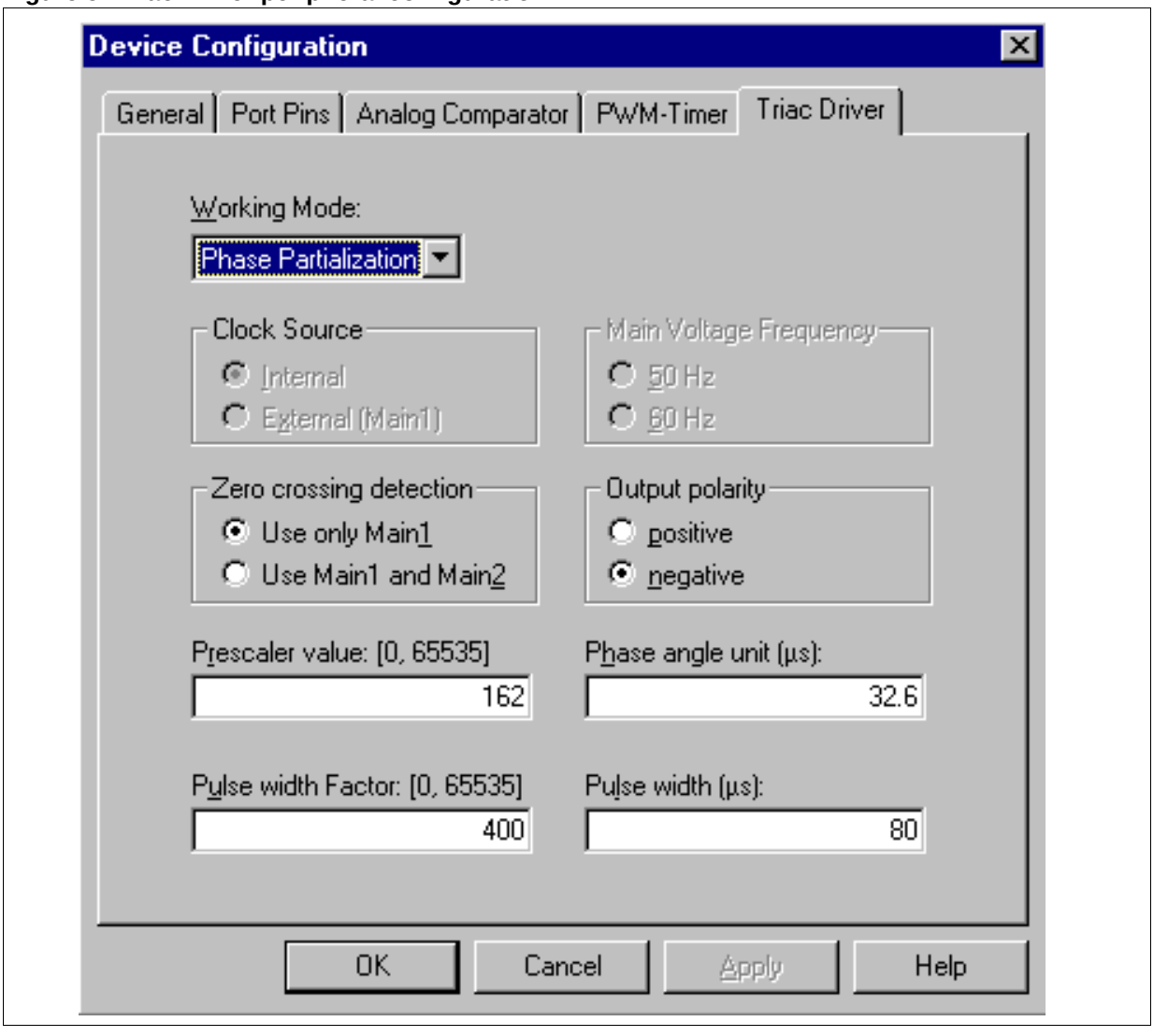
Pin PA0 configured as TROUT in order to have the TRIAC_DRIVER peripheral output on this pin (fig. 4).

Figure 4 : Port pins configuration.



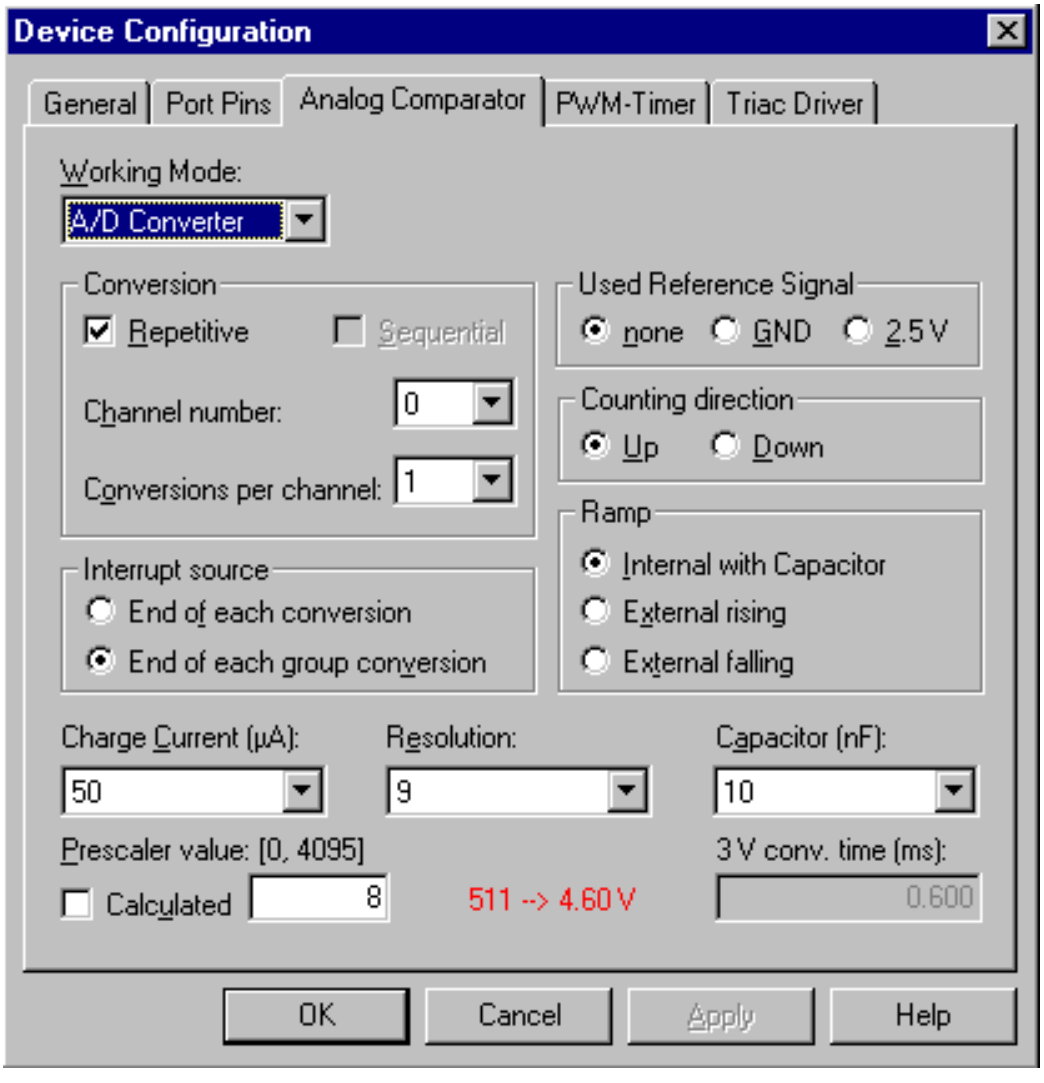
Pin PB0 configured as analog input Ain0. This also requires pin PB7 to be configured as CS (refer to ST52x440 datasheet and to application note 1548 "High Resolution Single Slope Conversion with the analog comparator of the ST52x440" for further information on this issue).

Figure 5 :Triac Driver peripheral configuration



Triac Driver peripheral configured in Phase partialization working mode (fig. 5), with firing pulse temporal width set at 80 us, control signal polarity set as negative and the phase angle unit fixed as 32.6 us: in this way every unitary variation in the TRIAC_COUNT register causes a variation of 32.6usec in the delay from mains voltage zero crossing to Triac firing.

Figure 6 : Analog Comparator configuration

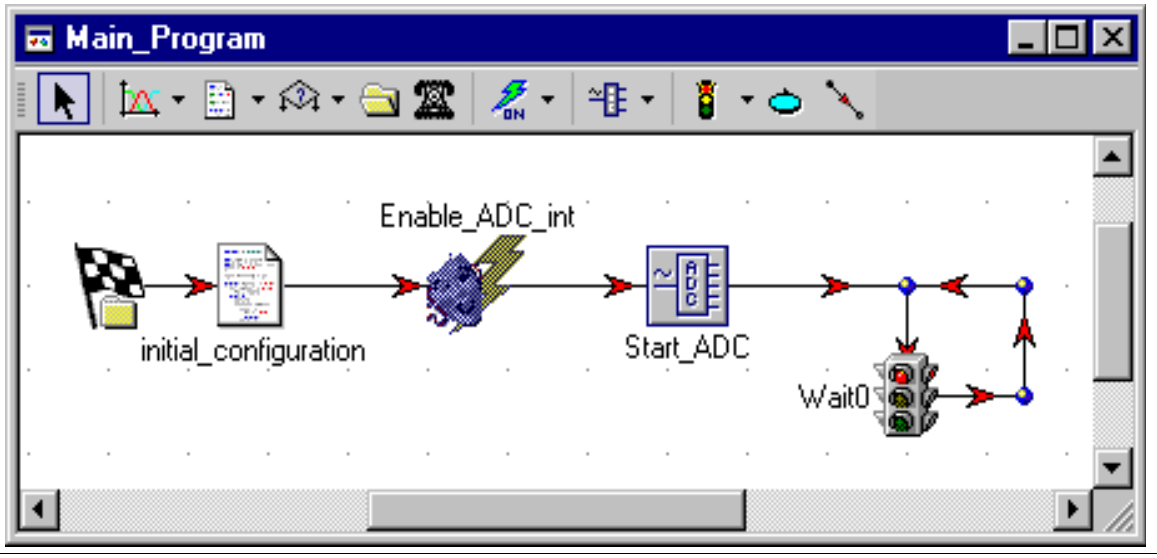


Analog comparator peripheral configured in A/D converter working mode (fig. 6). In this case an external capacitor is mandatory to generate the ramp to implement the AD conversion. The user can decide the slew of the ramp by setting the current generator value and the capacitor value. The peripheral converts continously the signal present at PB0/Ain0 input and provides an interrupt request at the end of every conversion. To have further information about the ST52x440 AD converter, please, refer to AN 1548.

5.2 Main Program

It enables interrupts from ADC and zero crossing detection, start ADC and enter wait state. From now on, only ADC interrupts wake up the core from wait state (figure 7).

Figure 7 : Main program.



5.3 ADC Interrupt Service Routine

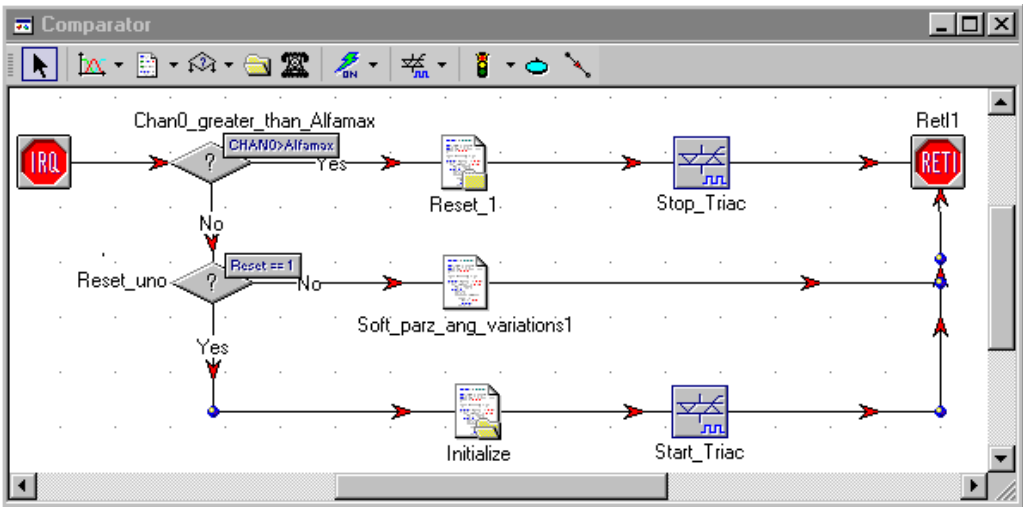
ADC continuously converts the value present at Ain0 input and provides an interrupt at every end of conversion.

ST52x440 updates the value of TRIAC_COUNT register - based on the value read by the ADC - in order to change the firing angle (in fig. 8 for the relative interrupt service routine)

In general, the analogue value converted by ADC is directly stored in the TRIAC_COUNT register. Notice that increasing the value present at Ain0, the firing angle also increases and the power decreases.

If the ADC reads a value greater than then Alfamax variable, the peripheral stops and no firing pulse is provided to the Triac gate. In fact, for firing angle corresponding to values of TRIAC_COUNT register greater than Alfamax, a small current flows on the motor, but the power is too low in order to run the motor. Obviously, this causes useless power losses on the system.

Figure 8 ADC Interrupt service routine.



When ADC converts a value greater than Alfamax the peripheral stops and flag variable Reset is used to memorize it. When ADC reads a value less than Alfamax the peripheral is enabled and on next AD conversions a soft start routine is implemented.

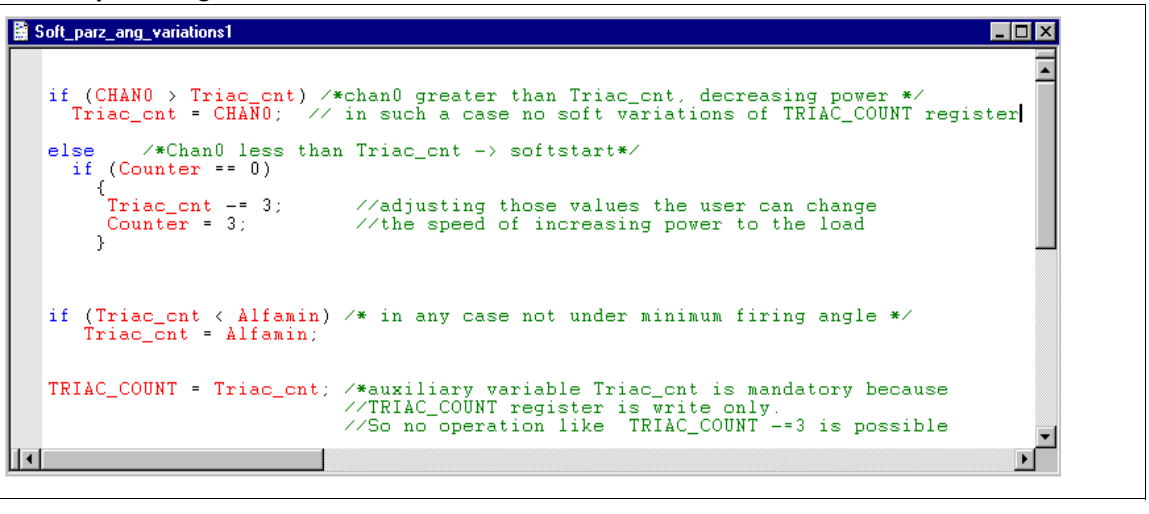
If the ADC reads a value less than the Alfamin variable, the TRIAC_COUNT register is loaded with the value Alfamin (fig. 9). In fact, if the firing angle is too low (corresponding to TRIAC_COUNT register values less than Alfamin), in presence of inductive loads like motors, the firing pulses end before the zero crossing of the current in the Triac, and so no Triac firing occurs. This is due to the delay existing between current and voltage waveforms on inductive loads (current delayed with respect to the voltage).

Both the Alfamin and Alfamax variables are to be adjusted according to experimental tests on the particular motor the user is going to use.

For resistive loads (like lamps for example) those precautions are not necessary (the user can have Alfamax = 255 and Alfamin = 0).

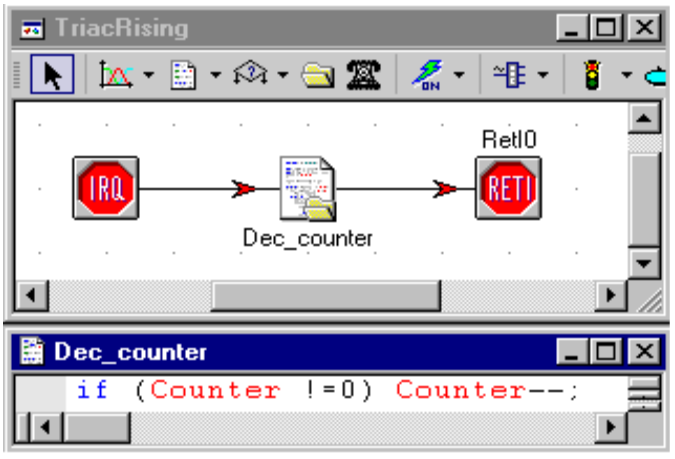
A soft start routine is also implemented. When the value read by the ADC is lower than TRIAC_COUNT register current value (i.e. the potentiometer has been rotated in order to increase the power delivered to the load), the TRIAC_COUNT register value is decremented by three only if the Counter variable is zero (fig. 9). This variable is loaded with three at each decrementation of the TRIAC_COUNT register value and then is decremented at each rising edge of the zero-crossing signal (every 20ms) until it reaches zero (fig.10). In this way the Counter reaches zero after 60ms, therefore the value stored in TRIAC_COUNT register is reduced once in 60ms. Adjusting the initial value of Counter variable and the TRIAC_COUNT value reduction step, the behavior of the soft start routine can be changed.

Figure 9 Soft_parz_ang_variations1 block.



Normally the value read from ADC is stored in the Triac_cnt temporary variable. If the user wants to increment the power on the load, a soft start routine is implemented. Triac_cnt is used because TRIAC_COUNT register is write-only.

Figure 10 Zero crossing rising edge interrupt service routine.



Every 20msecs the variable Counter is decremented if not set to zero. This variable reaches zero after 60msecs.

6 EXPERIMENTAL RESULTS

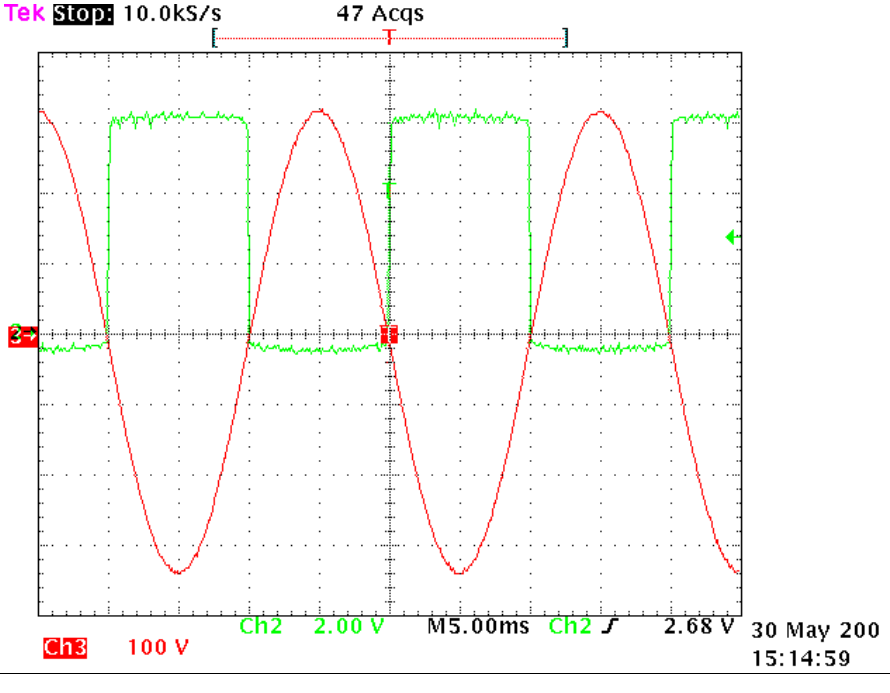
In this section we want to point out some experimental results obtained by using the circuit described in this application note to control a universal motor. Some oscilloscope pictures captured during laboratory tests are shown.

6.1 Zero Crossing Detection.

Figure 11 shows the mains voltage waveform and the relative zero crossing signal. In particular, the zero crossing signal is detected on pin 16 (PA1/Main1) of the micro: it is the Main1 input of the Triac Driver peripheral.

In order to obtain this signal, the micro controller pin internal protection diodes are used. In this manner, only a resistor can be used to constitute the zero crossing detection network.

Figure 11 Mains voltage waveform and relative zero crossing signal.



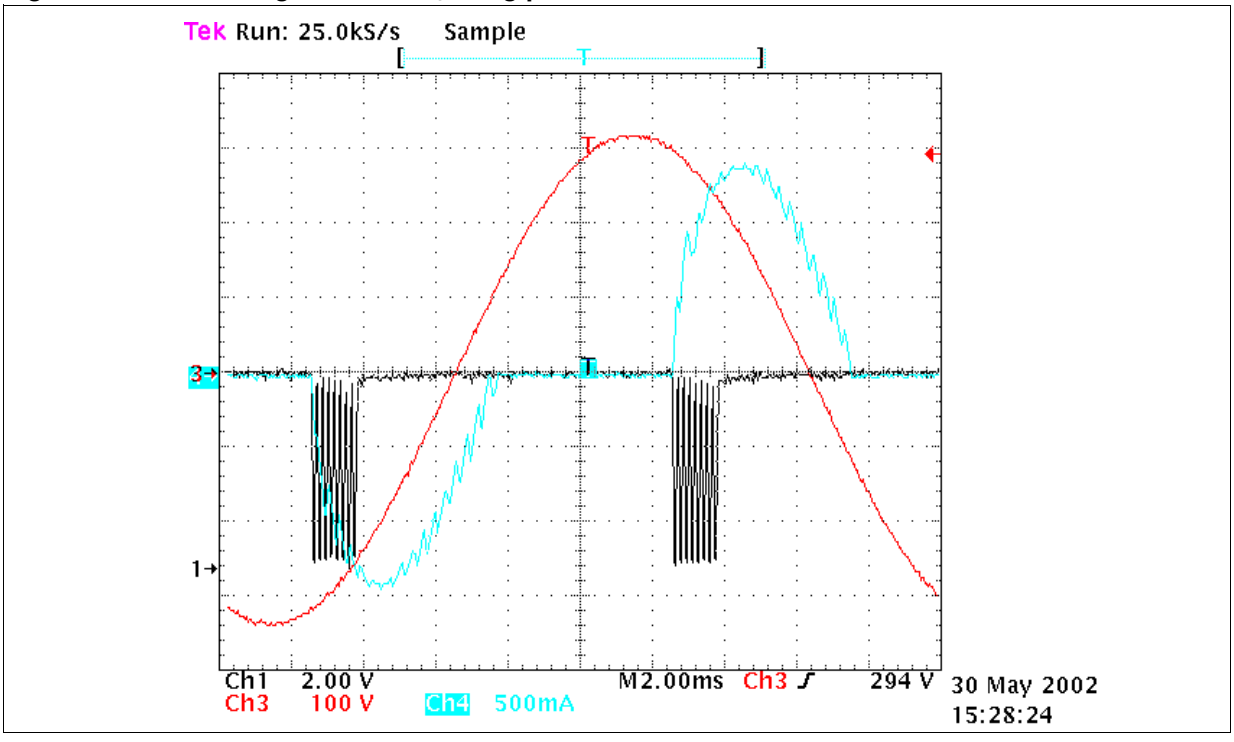
6.2 Current on the Load.

Fig.12 shows the mains voltage waveform, firing pulses for the Triac and the current on the load universal motor (the firing angle is approximately 100°). Note the delay to Triac firing.

The current waveform differs from the one shown on fig.2 because in this case the load is inductive, and so the current is delayed with respect to the voltage. For the same reason the current reaches zero after the zero crossing of the mains voltage. The noise on the current waveform, similar to a saw-tooth signal superimposed on it, is due to the motor brushes.

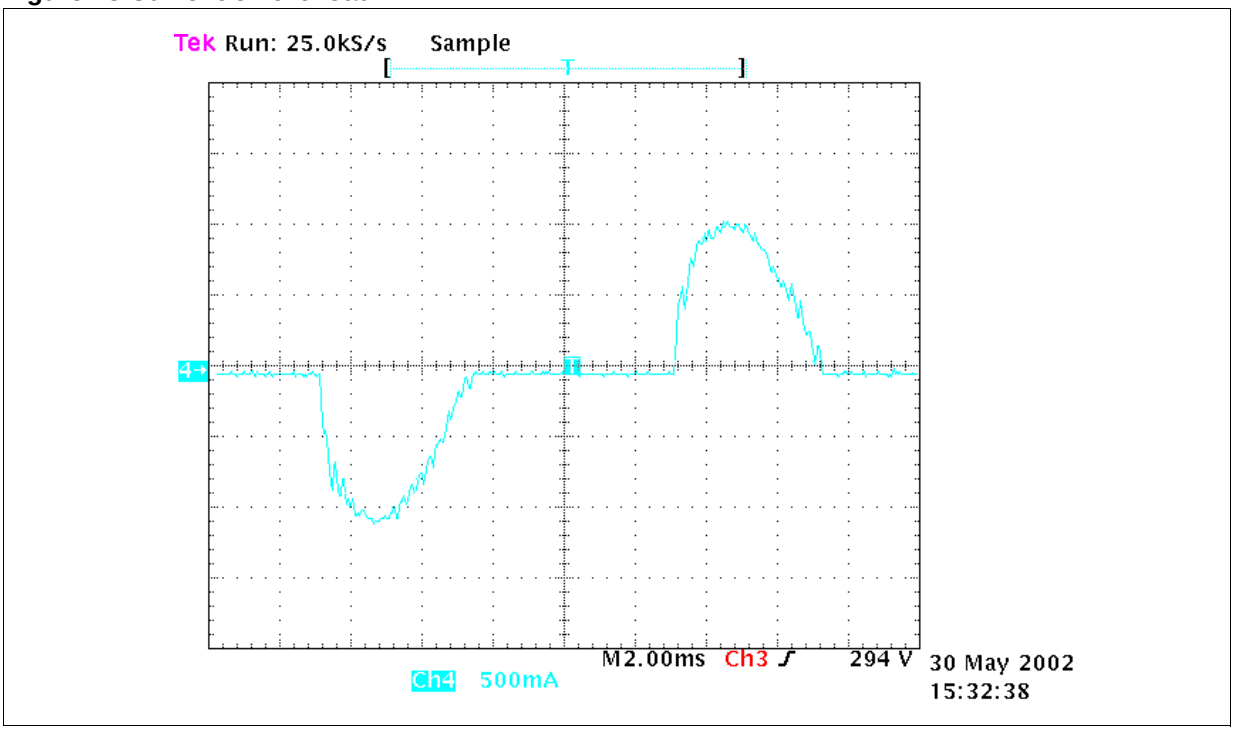
Fig.13 only shows the current waveform in this situation, typical of Phase Angle Partialization. This current waveform, very discontinuous with pulses on it, is the reason of very high order harmonics generated and of EMC problems related to this kind of motor control.

Figure 12 Mains voltage waveform, firing pulses and current on the load.



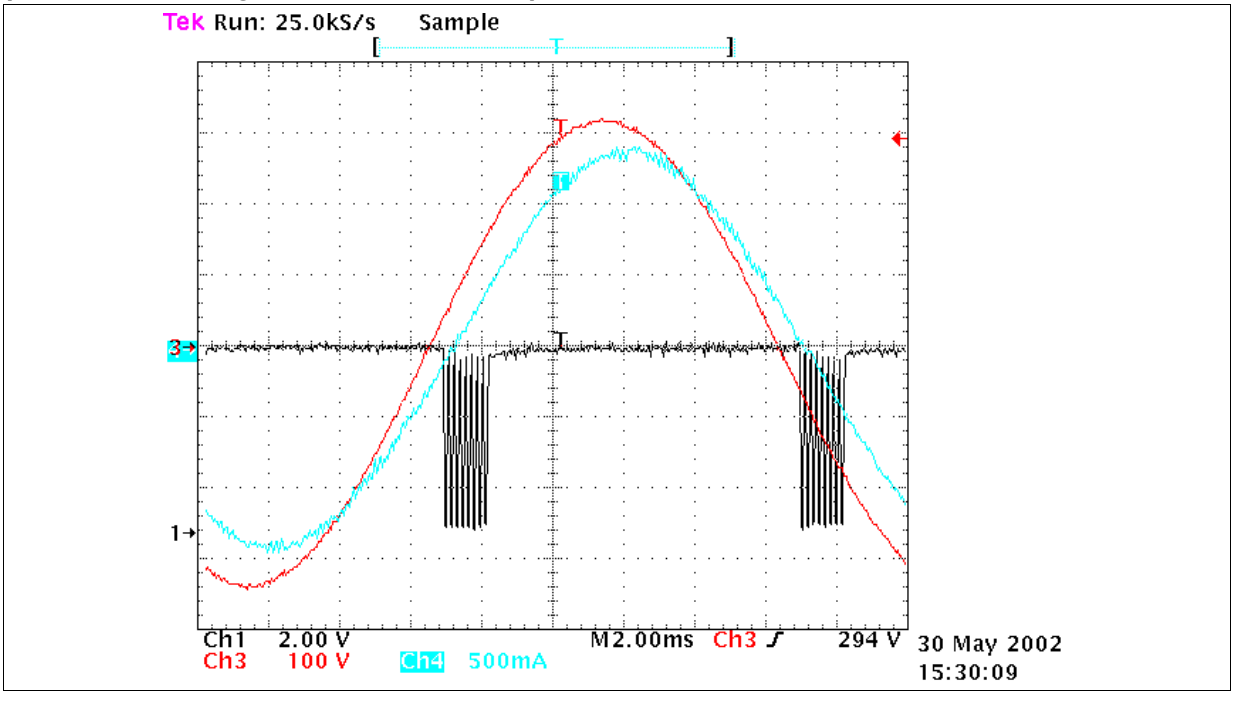
In the picture above, the Firing angle is approximately 100° .

Figure 13 Current on the load.



This waveform contains high order harmonics, which causes EMC problems for this kind of control. The firing angle is approximately 100°.

Figure 14 Mains voltage waveform, firing pulses and current on the load at full power (TRIAC_COUNT register contains Alfamin).



In figure 14 the current is sinusoidal. No high order harmonics are generated in this case. The picture shows the waveforms at full power rate: in this case, the conduction angle is 180° and the current on the load is sinusoidal, therefore EMC problems are not generated.

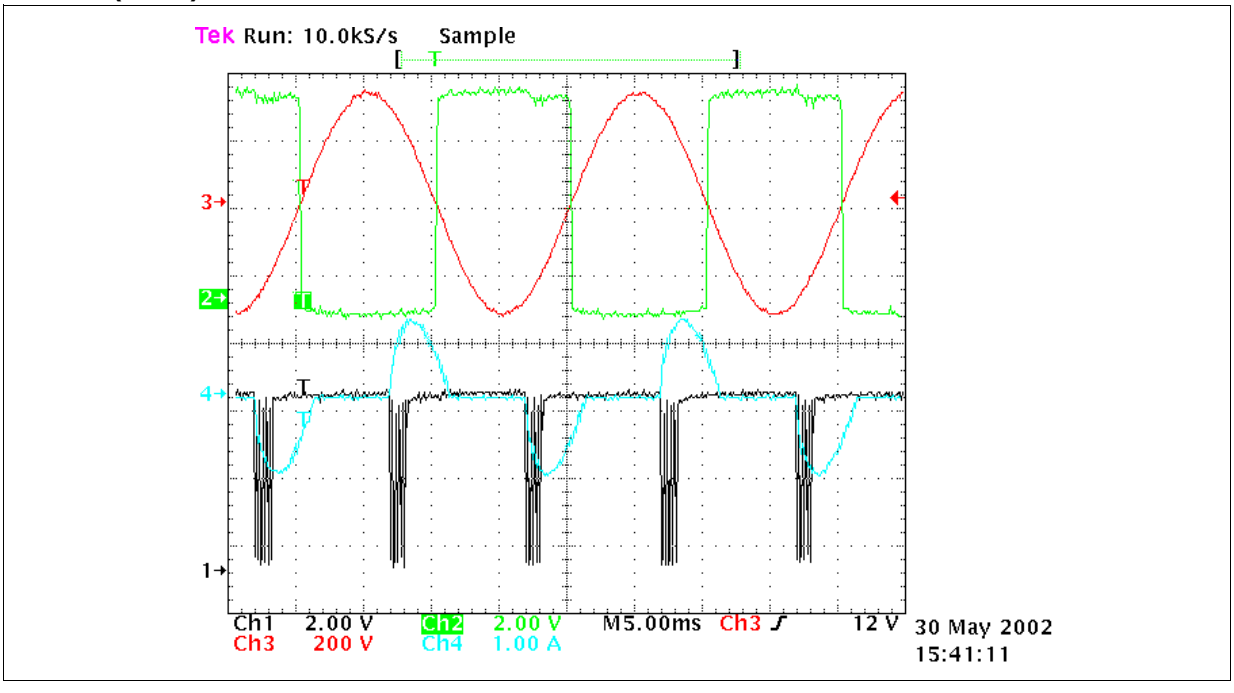
6.3 All Signals

Fig.15 shows the mains voltage waveform, relative zero-crossing signal, firing pulses for the Triac and the current on the motor (firing angle is approximately 100°).

In particular, on top of the picture, the mains voltage waveform and the relative zero-crossing signal are shown: the zero-crossing signal has a rising or falling edge on every zero crossing of the mains.

The bottom of the same picture shows the firing pulses for the Triac and the relative current waveform: after the Triac switches on, the current increases slowly due to the presence of an inductive load (the motor). For the same reason the current reaches zero after the zero crossing of the mains voltage (fig.12).

Figure 15 Mains voltage waveform, relative zero-crossing signal (up), firing pulses and current on the load (down).



Firing angle is approximately 100°.

7 ASSEMBLER CODE

Below follows the assembler code corresponding to the software described in the paragraphs above, as generated by Visual Five.

```
;  Interrupt Vector Configuration
   irq   0 Comparator
   irq   1 PwmTimer
   irq   2 TriacFalling
   irq   3 TriacRising
   irq   4 TriacPulse
   irq   5 External

;  Store Device Configuration Parameters into Eprom

;  IrqPriority (Hi-->Lo): Comparator PwmTimer TriacFalling TriacRising TriacPulse
   data  0   39  70 ;   RegConf_18 = 01000110
   data  0   38  136;  RegConf_17 = 10001000
;       RegConf_18_17 = 0100011010001000

;    Pin Configuration
   data  0   25  254;  RegConf_04 = 11111110
   data  0   32  63 ;   RegConf_11 = 00111111
   data  0   33  254;  RegConf_12 = 11111110
   data  0   34  255;  RegConf_13 = 11111111
   data  0   35  129;  RegConf_14 = 10000001

;    WatchDog Configuration
   data  0   23  15 ;  RegConf_02 = 00001111

;    BrownOut Configuration
   data  0   21  0  ;  RegConf_00 = 00000000
;    Comparator Configuration
   data  0   22  2  ;    RegConf_01 = 00000010
   data  0   24  177;  RegConf_03 = 10110001
   data  0   36  8  ;    RegConf_15 = 00001000
   data  0   37  0  ;    RegConf_16 = 00000000

;    Pwm-Timer Configuration
   data  0   26  0  ;   RegConf_05 = 00000000
   data  0   27  32 ;  RegConf_06 = 00100000
   data  0   28  0  ;   RegConf_07 = 00000000

;    Triac Driver Configuration
   data  0   31  73 ;   RegConf_10 = 01001001
;     Prescaler  = 162   //RegConf_08_09
   data  0   29  0  ;    RegConf_08 = 00000000
   data  0   30  162;  RegConf_09 = 10100010
```

```
;     PulseWidth = 400   //RegConf_19_20
    data  0   40  1  ;    RegConf_19 = 00000001
    data  0   41  144;  RegConf_20 = 10010000

    setmem 0   42
;  End ********************************************

;  Set Device Configuration Parameters
    pgset  0
    ldce  0   21  ;  RegConf_00 = 00000000
    ldce  1   22  ;  RegConf_01 = 00000010
    ldce  2   23  ;  RegConf_02 = 00001111
    ldce  3   24  ;  RegConf_03 = 10110001
    ldce  4   25  ;  RegConf_04 = 11111110
    ldce  5   26  ;  RegConf_05 = 00000000
    ldce  6   27  ;  RegConf_06 = 00100000
    ldce  7   28  ;  RegConf_07 = 00000000
    ldce  8   29  ;  RegConf_08 = 00000000
    ldce  9   30  ;   RegConf_09 = 10100010
    ldce  10  31  ;  RegConf_10 = 01001001
    ldce  11  32  ;  RegConf_11 = 00111111
    ldce  12  33  ;  RegConf_12 = 11111110
    ldce  13  34  ;  RegConf_13 = 11111111
    ldce  14  35  ;  RegConf_14 = 10000001
    ldce  15  36  ;  RegConf_15 = 00001000
    ldce  16  37  ;  RegConf_16 = 00000000
    ldce  17  38  ;  RegConf_17 = 10001000
    ldce  18  39  ;  RegConf_18 = 01000110
    ldce  19  40  ;  RegConf_19 = 00000001
    ldce  20  41  ;  RegConf_20 = 10010000

;       PwmTimer Preload = 0
    ldrc  0   0   ;  00000000b, 0x00
    ldpr  4   0


;  ********* User Defined Variables *********
;       NAME -> REG
;  -----------------------------------------
;    Alfamax ->   0
;    Alfamin ->   1
;    Counter ->   2
;       Reset ->   3
;  Triac_cnt ->   4
;  *****************************************
main:
;  ********** Start procedure "main"
```

```
Start:

conf_iniziale:
   ldrc  1   15  ;  00001111b, 0x0F
   ldrc  0   225 ;  11100001b, 0xE1
   ldrc  3   1   ;  00000001b, 0x01

IrqMask_0:

;  IrqEnableMask
;     Enable: Comparator TriacRising
;    Disable: PwmTimer TriacFalling TriacPulse External
;    External Interrupt Polarity: RISING
   ldrc  5   18
   ldcr  0   5   ;  RegConf_00 = 00010010

Start_AD:
;  Comparator Setting
   ldrc  5   1
   ldcr  16  5   ;  RegConf_16 = 00000001
   ldrc  5   177
   ldcr  3   5   ;  RegConf_03 = 10110001
   ldrc  5   2
   ldcr  1   5   ;  RegConf_01 = 00000010

Wait0:
   waiti
   jp    Wait0
External:
;  ********** Start procedure "External"

RetI2:
   reti

TriacPulse:
;  ********** Start procedure "TriacPulse"

RetI5:
   reti


TriacRising:
;  ********** Start procedure "TriacRising"

Dec_counter:
   ldrc  6   0   ;  00000000b, 0x00
   sub   6   2
```

```
   jpz   End_If
   dec   2

End_If:

RetI0:
   reti


TriacFalling:
;  ********** Start procedure "TriacFalling"

RetI4:
   reti


PwmTimer:
;  ********** Start procedure "PwmTimer"

RetI3:
   reti


Comparator:
;  ********** Start procedure "Comparator"

Chan0_supera_Alfamax:
   ldri  7   1
   ldri  8   2
   ldrc  9   0   ;  00000000b, 0x00
   sub   9   7
   jpnz  label
   ldrr  9   0
   sub   9   8
   jpns  End_If_1

label:
   jp    Reset_a_1

End_If_1:
Reset_uno:
   ldrc  7   1   ;  00000001b, 0x01
   sub   7   3
   jpnz  End_If_2
   jp    Initialize_start_Triac

End_If_2:
```

```
Soft_parz_ang_variations1:
   ldri  7   1
   ldri  8   2
   ldrc  9   0   ;  00000000b, 0x00
   sub   9   7
   jpnz  label_1
   ldrr  9   4
   sub   9   8
   jpns  No_If_4

label_1:
   ldri  4   2
   jp    End_If_4

No_If_4:
   ldrc  7   0   ;  00000000b, 0x00
   sub   7   2
   jpnz  End_If_3
   ldrc  7   3   ;  00000011b, 0x03
   sub   4   7
   ldrc  2   3   ;  00000011b, 0x03

End_If_3:
End_If_4:
   ldrr  7   4
   sub   7   1
   jpns  End_If_5
   ldrr  4   1

End_If_5:
   ldpr  9   4

RetI1:
   reti

Initialize_start_Triac:
   ldrc  3   0   ;  00000000b, 0x00
   ldpr  9   0
   ldrr  4   0
   ldrc  2   3   ;  00000011b, 0x03

Triac_Driver_0:
;  TriacDriver Setting
   ldrc  7   233
   ldcr  10  7   ;  RegConf_10 = 11101001
   jp    RetI1
```

```
Reset_a_1:
   ldrc  3   1   ;  00000001b, 0x01


stop_triac:
;  TriacDriver Setting
   ldrc  7   73
   ldcr  10  7   ;  RegConf_10 = 01001001
   jp    RetI1
```

h. APPENDIX A.

Figures 16 and 17 show the PCB, Assembly Top and a picture of the board described in this application note.

Figure 16 :PCB of the circuit described in this application note.The scale ratio is not 1:1

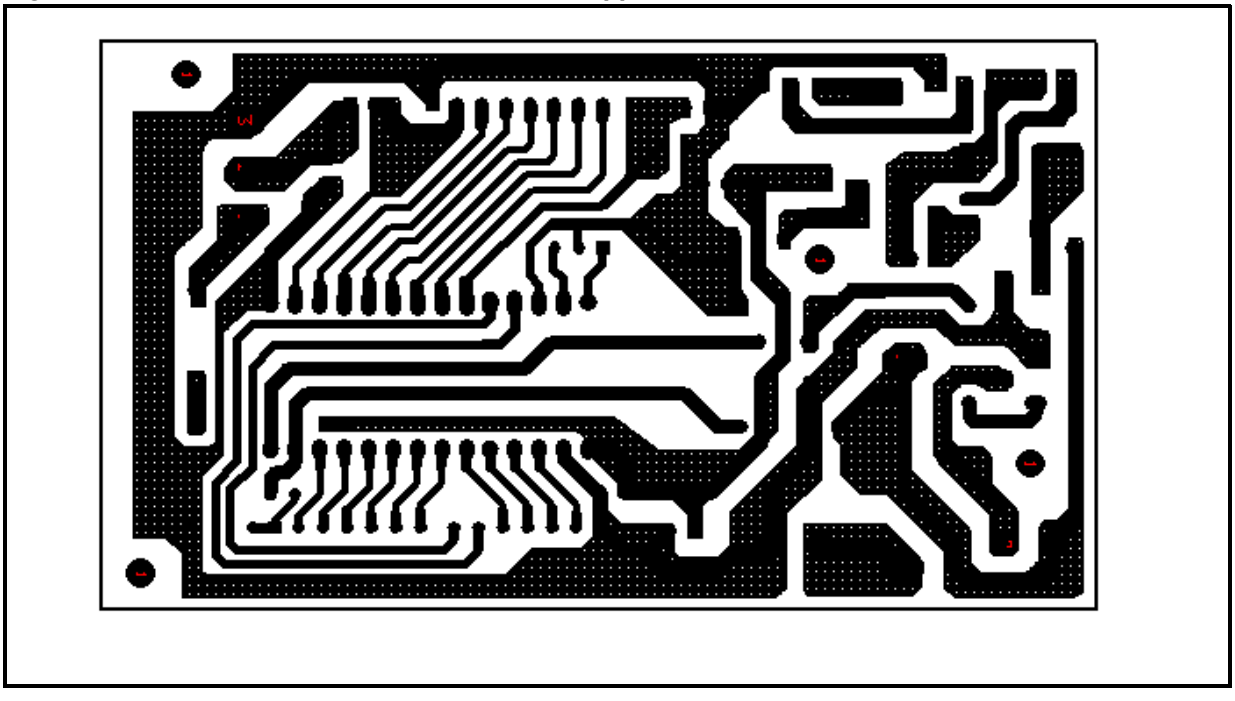
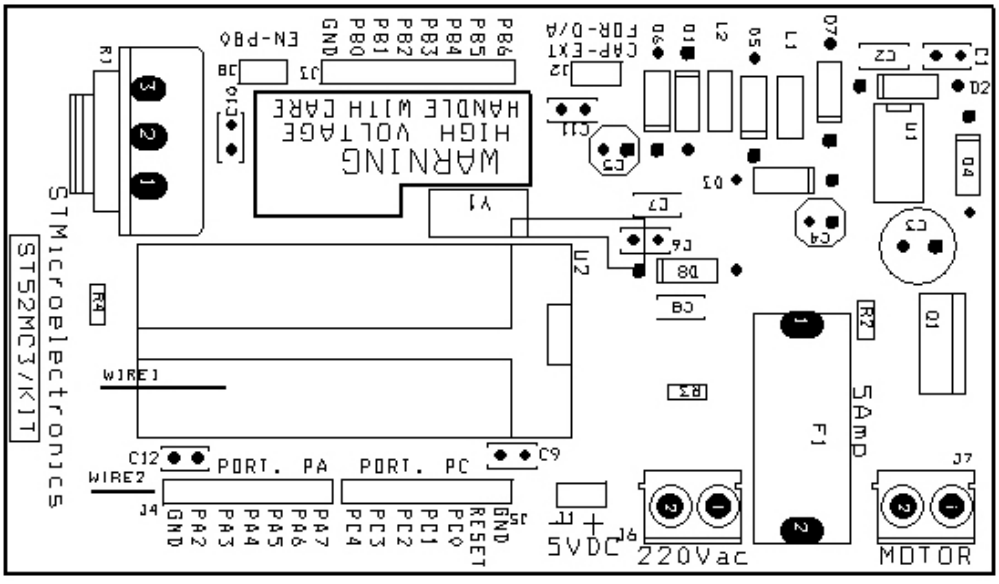


Figure 17 Assembly Top of the circuit described in this application note.



REFERENCES

[1] - ST52x440 datasheet: 8 - BIT ICU WITH TIMER/PWM, ANALOG COMPARATOR, TRIAC/PWM TIMER, WDG, UP TO 8K EPROM.

[2] - VISUAL FIVE 5.0 USER MANUAL.

[3] -AN1548: HIGH RESOLUTION SINGLE SLOPE CONVERSION WITH THE ANALOG COMPARATOR OF THE ST52X440.

[4] - AN1221: A SIMPLE METHOD TO DRIVE A TRIAC WITH ST52X420 DIRECTLY FROM THE MAIN.

[5] - AN1295: A DEVELOPMENT BOARD TO DRIVE A TRIAC WITH ST52X420 DIRECTLY FROM THE MAINS.

BOM

ST52MC3/KIT Revised: Friday, February 22, 2002
Revision:
Bill Of Materials February 25,2002 12:26:19 Page1

Item	Quantity	Reference	Part	cod.RS/Elcart
1	1	C1	47nF 50V	211-5514
2	1	C2	470nF 50V	211-5570
3	1	C3	1uF 450V	228-7142
4	1	C4	10uF 25V	116-868
5	1	C5	47uF 35V	116-947
6	2	C6,C9	100nF 50V	211-5491
7	1	C7	.1uF/63V	179-4396
8	2	C8,C11	10nF 50V	211-5514
9	1	C10	100pF 50V	188-6079
9a	1	C12	1/8.2nF	OPTIONAL, NOT USED
10	2	D1,D7	STTA106	STM
11	1	D2	BZX85C5V1	348-6148
12	2	D3,D4	BYT11/600	STM
13	1	D5	BYV 100/200	STM
14	1	D6	BZX85C6V8	812-421
15	1	D8	BYV10-40	STM
16	1	F1	5A	4/10505+4/100
36				
17	1	2PIN J1	5VDC-EXT	5/7646
18	1	2PIN J2	CAP. FOR D/A CONV.	5/7646
19	1	8PIN J3	PORT PB	5/7646
20	1	7PIN J4	PORT PA	5/7646
21	1	6PIN J5	PORT PC	5/7646
22	1	2PIN J6	220Vac-input	5/5130
23	1	2PIN J7	Motor	5/5130
24	1	2PIN J8	EN-PB0	5/7646
25	1	L1	1.8mH	191-0734
26	1	L2	470uH	191-0677
27	1	Q1	BTB16-600CW	STM
28	1	R1	47K-A type potentiometer LPHA	2/1230
29	1	R2	100 5%	
30	1	R3	330K 5%	
31	1	R4	47K 5%	
32	1	U1	VIPer12ADIP	STM
33	1	U2	ST52x440	801-768
34	1	Y1	4.91Mhz	179-3747
35	2	WIRE1/2	CAVALLOTTI	fixed wire L.14mm
36	4		tacky pins.	