



Using the ST72141 Motor Control MCU in Sensor Mode

by 8-bit Microcontroller Application Lab

INTRODUCTION

The ST72141K microcontroller is especially designed for Brushless DC Motor Control applications due to its following features:

- Motor Control Unit (MTC), that can be seen as a Pulse Width Modulator multiplex on six output channels,
- Back Electromotive Force (B-EMF) Zero-Crossing-Detector for sensorless control of permanent magnet direct current brushless motors.

The MTC is designed for sensorless control but can also easily support the use of sensors (generally, Hall-effect sensors) which may be useful for very low-speed or positioning applications. The purpose of this Application Note is to describe how to initialize the motor control unit in order to run a given motor equipped with Hall-effect sensors.

This application note is divided into different segments with code examples given in Assembly language.

Table of Contents

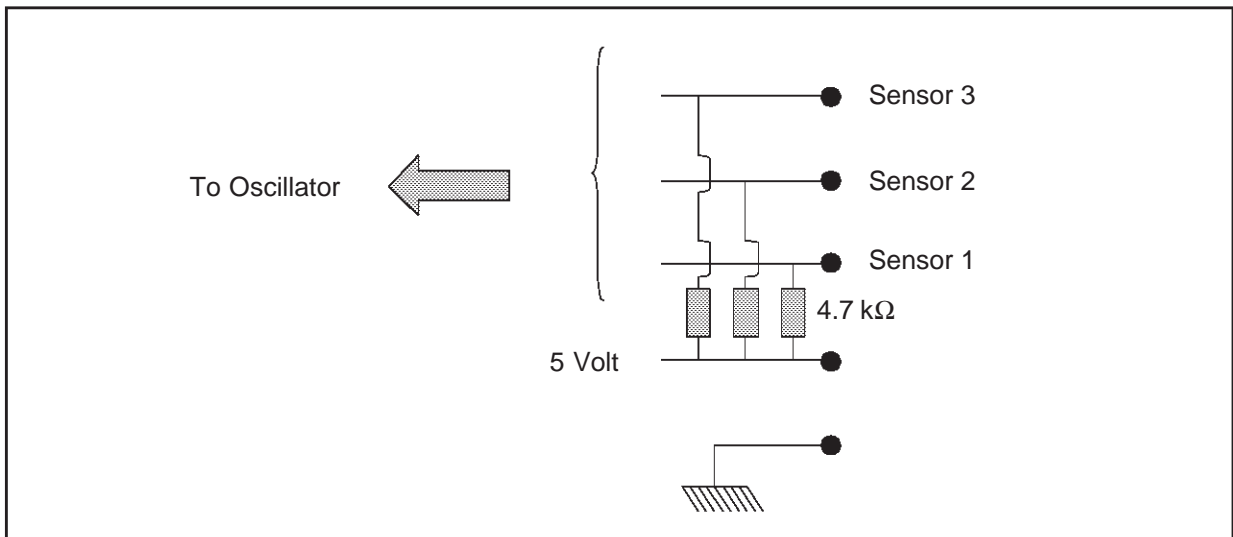
INTRODUCTION	1
1 PHASE COMMUTATION INITIALIZATION	3
1.1 SENSOR BEHAVIOUR	3
1.2 CONNECTING THE SENSORS TO THE MICROCONTROLLER	5
1.3 PHASE COMMUTATION TABLE DECLARATION	5
2 STARTING IN SENSOR MODE	6
2.1 ROTOR POSITION	6
2.2 OUTPUT THE FIRST PHASE STEP	8
3 INTERRUPT SERVICE ROUTINES	10
3.1 EMERGENCY AND OVERFLOW INTERRUPTS	10
3.2 COMMUTATION AND DEMAGNETIZATION INTERRUPTS	10
3.3 RATIO AND ZERO-CROSSING INTERRUPTS	10
4 VOLTAGE/CURRENT MODE	10
5 SENSOR MODE WITH THE MOTOR CONTROL KANDA KIT VERSION 2	11
5.1 SENSOR WAVEFORM	11
5.2 HARDWARE MODIFICATIONS	12
5.3 SOFTWARE MODIFICATIONS	13
5.3.1 Modifications in the Constant Declaration File (.def source file)	13
5.3.2 Modifications in the Main File (.stm source file)	14
5.3.3 Modifications to the Position_detection Sub-routine (.sts source file)	14
6 CONCLUSION	15

1 PHASE COMMUTATION INITIALIZATION

1.1 SENSOR BEHAVIOUR

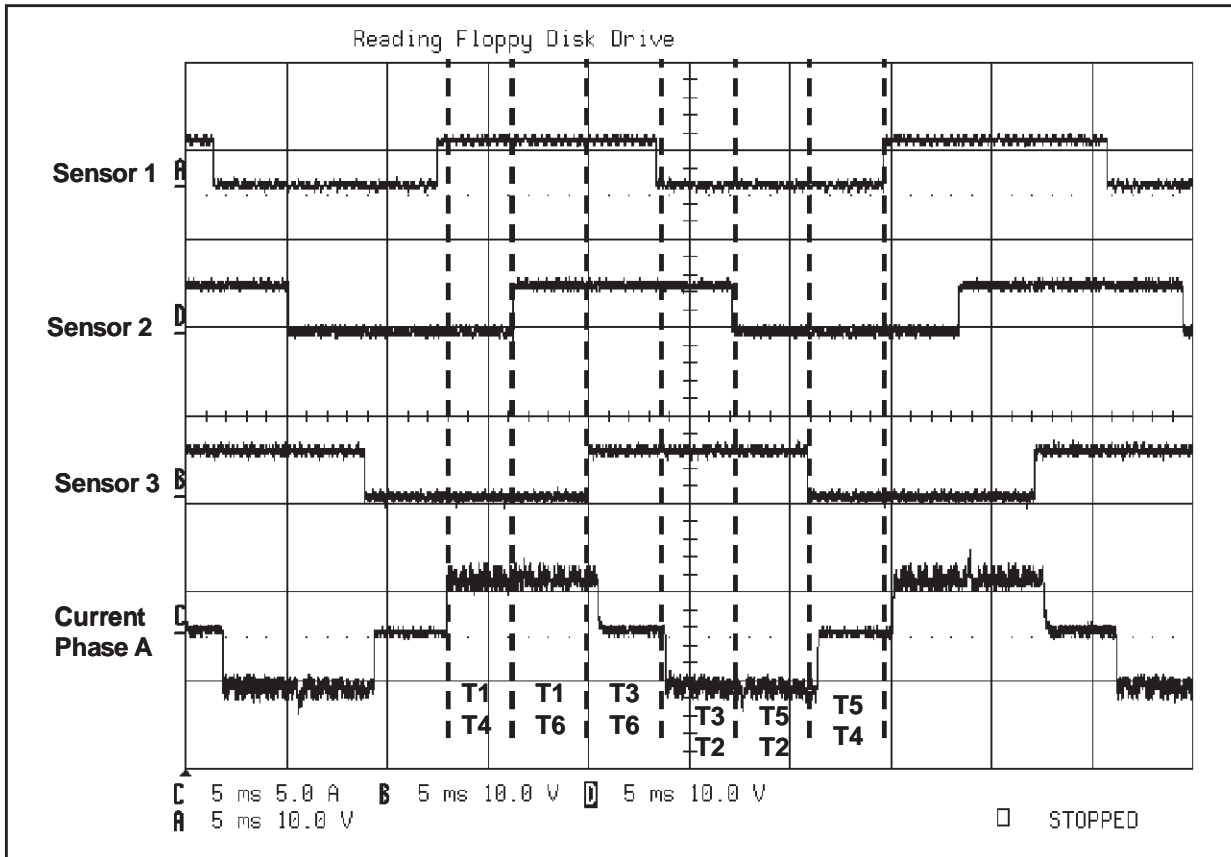
In order to make your motor run in Sensor mode, the first thing you need to determine is how the 3 sensors behave in front of the 3 motor phases. To do this, the sensors must be connected to a power supply (generally, 5 V) and the waveforms of the 3 sensors are observed while the motor is running in Sensorless mode using an oscilloscope, for example, or when rotating using an external tool. The oscilloscope is used to display the behaviour of the sensors during the 6-Step Principle used to drive the motor and for the transition between each of the steps. Figure 1 shows how to connect the 3 sensors to an oscilloscope in order to observe their behaviour.

Figure 1. Oscilloscope Set-up



Once the sensors are connected as shown above, the waveforms can be observed on the oscilloscope. Figure 2 is a screen shot of the oscilloscope display taken while the motor was running in Sensorless mode. The motor used in this example is the Ametek motor included in the Kanda Motor Control Kit (ST7MTC2).

Figure 2. Oscilloscope Screen Shot



Several different waveforms must be displayed on the oscilloscope. First, the waveforms of each of the 3 sensors. Second, the current on one phase of the motor must be displayed in order to verify how the sensors behave in relation to each step of the 6-Step Principle used to drive the motor.

Note: The 6-Step Principle is described in Application Note 1130.

The information in the table below is determined by this screen shot.

Table 1. Sensor State and Transition Table

Step	Sensor State			Transition between current step and next step
	Sensor 1	Sensor 2	Sensor 3	
T1-T4	1	0	0	Rising edge sensor 2
T1-T6	1	1	0	Rising edge sensor 3
T3-T6	1	1	1	Falling edge sensor 1
T3-T2	0	1	1	Falling edge sensor 2
T5-T2	0	0	1	Falling edge sensor 3
T5-T4	0	0	0	Rising edge sensor 1

Warning: When the sensor behaviour capture has been done in Sensorless mode with the capture of the current in phase as shown in Figure 2, all Sensor mode transition and commutation tables will be

written using this capture. Therefore, the motor phase connections must not be modified after this capture, otherwise they will no longer correspond to the specific state.

1.2 CONNECTING THE SENSORS TO THE MICROCONTROLLER

After the behaviour of the sensors has been determined, they must be connected to the ST72141 Motor Control MCU. The user must select which of the 3 comparators will be connected to each sensor. In this example, the following configuration is used:

- Sensor 3 is connected to MCIA
- Sensor 2 is connected to MCIB
- Sensor 1 is connected to MCIC

1.3 PHASE COMMUTATION TABLE DECLARATION

Now, we have all the information needed to write the phase commutation tables corresponding to Sensor mode. There are 3 phase commutation tables that correspond to the following information:

- the configuration of the MCIA, MCIB and MCIC comparators for each of the 6 steps,
- the configuration of the MCRB register for each step in Current driving mode,
- the configuration of the MCRB register for each step in Voltage driving mode.

Note: There is no need to write both of the tables for the MCRB register if only one of the driving modes is being used.

Each table consists of 6 lines that correspond to each of the 6 steps of the motor drive principle. The tables must be written in the constant declaration source file of your software program.

```
;table for sensorFULL motor
;THESE TABLES ARE FOR THE AMETEK MOTOR PROVIDED WITH THE KIT
    ;II
    ;SSTTTTTT
    ;10563412

    ;bbbbbbbb
    ;76543210

.phreg_SR
    DC.B %01000110 ;T1-T4 or T7-T10 Valid compareteur B (Hall 2)
    DC.B %00010010 ;T1-T6 or T7-T12 Valid compareteur A (Hall 3)
    DC.B %10011000 ;T3-T6 or T9-T12 Valid compareteur C (Hall 1)
    DC.B %01001001 ;T3-T2 or T9-T8 Valid compareteur B (Hall 2)
    DC.B %00100001 ;T5-T2 or T11-T8 Valid compareteur A (Hall 3)
    DC.B %10100100 ;T5-T4 or T11-T10 Valid compareteur C (Hall 1)
```

Using the ST72141 Motor Control MCU in Sensor Mode

The code above is an example of the first phase commutation table corresponding to the validation of the correct comparator for each of the 6 steps. Each byte of this table is stored in the MPHST register in order to validate the correct switch and comparator for each step.

Note: For more information about the Motor Control Peripheral Register (MCPR) register description, please refer to Application Note 1082.

In the above example, switches T1 and T4 are validated in step 1 (pins MCO1 and MCO2 of the microcontroller and bits 1 and 2 of the MPHST register, depending on your hardware configuration). Table 1 shows that the transition on this step is a rising edge on sensor 2. Since sensor 2 is connected to comparator MCIB of the microcontroller, comparator MCIB is validated with bits 6 and 7 of the MPHST register. This is the same principle for the other steps.

After this table is written, the transition on the sensor for each step (falling or rising edge) must be indicated in the software. This information is written in the MCRB register. A second table has to be written if both Voltage and Current driving modes are used. This transition information is written in the MCRB register (bit 5: CPB). An example of this table for Voltage mode is shown below.

```
    ; Voltage mode CHSO [CPB=1 0->1 !! CPB=0 1->0 is BEMF on selected phase]
    ; PDDS [OS2=0 not alternate, OS2=1 Alternate during de-
magnetisation]

    ; BMM2
    ; bbbbbbbb
    ; 76543210

.volt_crbreg_SR
    DC.B %00110000 ;CPB=1 Hall 2:0-->1
    DC.B %00111000 ;CPB=1 Hall 3:0-->1
    DC.B %00010000 ;CPB=0 Hall 1:1-->0
    DC.B %00011000 ;CPB=0 Hall 2:1-->0
    DC.B %00010000 ;CPB=0 Hall 3:1-->0
    DC.B %00111000 ;CPB=1 Hall 1:0-->1
```

In this example, all the transitions for the 6 steps are given in the CPB bit, which is set in step 1 (first line of the table). This means that the microcontroller will wait for the commutation signal on a rising edge transition on the sensor. The CPB bit is set because, as shown in Table 1, in step 1 the transition is a rising edge on sensor 2.

2 STARTING IN SENSOR MODE

2.1 ROTOR POSITION

Starting in Sensor mode enables the motor to start directly in Auto-switched mode. The data sent by the sensors is used to determine the rotor position. This data should be read in order

to prevent any unwanted switching of motor phases during the start-up phase. The pre-positioning phase, required for in Sensorless driving mode, is no longer needed in this case.

Note: For more information, please refer to Application Note 1276.

During the initialization phase, the motor control peripheral is reset and the status of the Emergency Input pin (MCES) is checked.

All three sensors inputs (MCIA, MCIB and MCIC) must be read sequentially in order to determine the rotor position. This is done by:

- selecting the signal to be read via bits IS[1:0] of the MPHST register,
- selecting the voltage reference via bits VR[1:0] of the MCRB register,
- setting Indirect Access mode in the phase state register (DAC),
- setting the output enable bit (MOE) in the MCRA register.

When switching to Direct Access mode, the sensor data is latched and the comparator output can be read via the HST bit in the MIMR register. This sequence must be repeated until all signals have been read. A NOP instruction is needed to read the comparator due to a conflict between the core clock and the peripheral clock. Adding a NOP instruction allows a good sampling of the comparator values.

Depending on the rotor position, the appropriate phases for the commutation event have to be set in the MPHST register. The rotor position is verified twice to prevent any errors.

```
.position_detection
    push  A
.pos_detect
    clr   count ;reset temporary mem
    clr   A
    ld    MPHST,A ;valid MCIA with all switches off
    bres  MCRA,#4 ;DAC bit
    nop
    bset  MCRA,#4
    ld    A,MIMR
    and   A,#$80 ;get level of A sensor
    jreq  get_B
    inc   count ;mem sensor A=1
.get_B
    ld    A,$40
    ld    MPHST,A ;valid MCIB with all switches off
    bres  MCRA,#4 ;DAC bit
    nop
    bset  MCRA,#4
    ld    A,MIMR
    and   A,$80 ;get level of B sensor
    jreq  get_C
    ld    A,count
```

```
        or    A, #02 ;mem sensor B=1
        ld    count, A
.get_C
        ld    A, #80
        ld    MPHST, A; valid MCIC with all switches off
        bres  MCRA, #4 ;DAC bit
        nop
        bset  MCRA, #4
        ld    A, MIMR
        and   A, #80 ;get level of C sensor
        jreq  verific_sensor
        ld    A, count
        or    A, #04 ;mem sensor C=1
        ld    count, A
```

2.2 OUTPUT THE FIRST PHASE STEP

After the rotor position detection routine, a value is memorised corresponding to the starting step of the motor. Using the known rotor position, the output phase configuration for the first commutation can be set. As no timing data is available, a commutation event will be initialized. Below is an example of code where the first phase configuration is set in the phasestep parameter.

```
.control_sensor
        ld    A, count
        cp    A, count1
        jreq  read_sens_ok ;position checked
        jp    pos_detect   ;check again the position
.read_sens_ok
        cp    A, #04
        jrne  step2
        clr   x          ;step T1-T4 or T7-T10
        jp    end_sens_detect
.step2
        cp    A, #06
        jrne  step3
        ld    x, #1      ;step T1-T6 or T7-T12
        jp    end_sens_detect
.step3
        cp    A, #07
        jrne  step4
        ld    x, #2      ;step T3-T6 or T9-T12
        jp    end_sens_detect
.step4
```



```
        cp    A, #03
        jrne  step5
        ld    x, #3      ;step T3-T2 or T9-T8
        jp    end_sens_detect
.step5
        cp    A, #01
        jrne  step6      ;step T5-T2 or T11-T8
        ld    x, #4
        jp    end_sens_detect
.step6
        cp    A, #00
        jrne  def_detect
        ld    x, #5      ;step T5-T4 or T11-T10
        jp    end_sens_detect
.def_detect
        jp    pos_detect
.end_sens_detect
        ld    phasestep, X
        pop   A
        ret
```

Depending on the motor characteristics, the appropriated values for the MPRSR and MCOMP registers, the PWM signal frequency and the duty cycle of Timer A must be initialized during the initialisation phase in the software, i.e. before any interrupts are enabled. The rotor position determined in Section 2.1 is the current position.

In order to start the motor without jerking, the next phase and zero-crossing event parameters must be set. Special care is needed when setting the next phase parameters. In this example, increment the phasestep parameter in order to enter the next step configuration. In Sensor mode, when the rotor position is read, the phasestep parameter can have a value between 0 and 5. After the phasestep parameter is incremented, verify that its value is less than 6. If the phasestep value is equal to 6 after incrementation, the parameter must be cleared to re-start with the first step.

Synchronous mode is selected in the MCRA register during the initialisation phase. In the software parameters, the event detection interrupt must be validated immediately and only one event is needed to pass in Auto-switched mode. This enables the timings to be automatically computed and written to the registers when the next commutation event is detected. At the end of the initialisation phase and before the main part of the software, the SWA bit can be set in the MCRA register to immediately enable Auto-switched mode after the initialisation phase.

If the PWM duty cycle is increased, the speed or torque of the motor will also increase depending on the selected driving mode.

3 INTERRUPT SERVICE ROUTINES

3.1 EMERGENCY AND OVERFLOW INTERRUPTS

In the event that an emergency interrupt is detected, the outputs should be set to high impedance and all MTC registers and timers should be reset.

This application note does not deal with multiplier overflow interrupts. For more information about this type of interrupt, please refer to the ST72141 Datasheet.

3.2 COMMUTATION AND DEMAGNETIZATION INTERRUPTS

The commutation interrupt service routine is only entered at the end of the first phase step. Afterwards, this routine is disabled in order to receive valid data for the MZREG and MZPRV registers for further motor control purposes.

Demagnetization interrupts are not enabled in Sensor mode

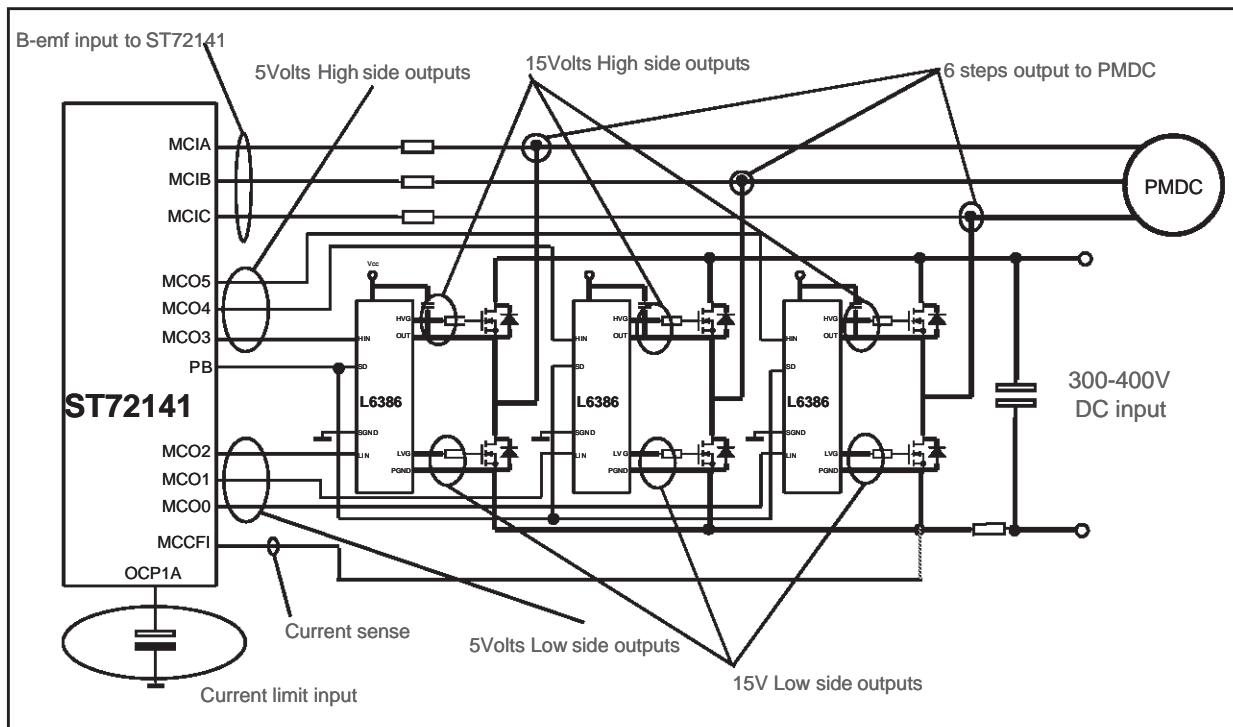
3.3 RATIO AND ZERO-CROSSING INTERRUPTS

Each time a zero-crossing (sensor signal change) is detected, an interrupt will occur. This will consecutively output the six phase configurations required to run the motor.

4 VOLTAGE/CURRENT MODE

The ST72141 Motor Control MCU can operate in two driving modes: Current mode and Voltage mode. The motor driving mode is selected via the VOC1 bit in the MCRA register (bit 3).

Figure 3. Hardware Configuration Example



For applications using a lower bridge voltage, refer to the L9931 power interface circuitry for 3-phase Brushless DC motor driving systems.

5 SENSOR MODE WITH THE MOTOR CONTROL KANDA KIT VERSION 2

The purpose of this section is to explain how to quickly pass into Sensor mode using Version 2 of the Kanda Motor Control Kit. This is possible in stage 3 of the kit. In the generic software used in stages 1 and 2, the Sensor mode routines are included. In stage 3 of the kit, once the software program is edited and the parameters saved, certain minor hardware and software modifications are needed in order for the motor to switch into Sensor mode.

5.1 SENSOR WAVEFORM

As explained in Section 1, the sensor waveforms of the motor must first be displayed in order to determine the values for the transition table (see Table 1).

Once the values for the transition table and the correct parameters for the motor are determined, save the software configuration and pass to stage 3. Run the application builder in stage 3 in order to edit and access the software program. When the program is completed, the microcontroller board and the software must be modified before being able to pass into Sensor mode.

Warning: Do not forget, as explained in Section 1, that once the sensor behaviour capture has been de-

Using the ST72141 Motor Control MCU in Sensor Mode

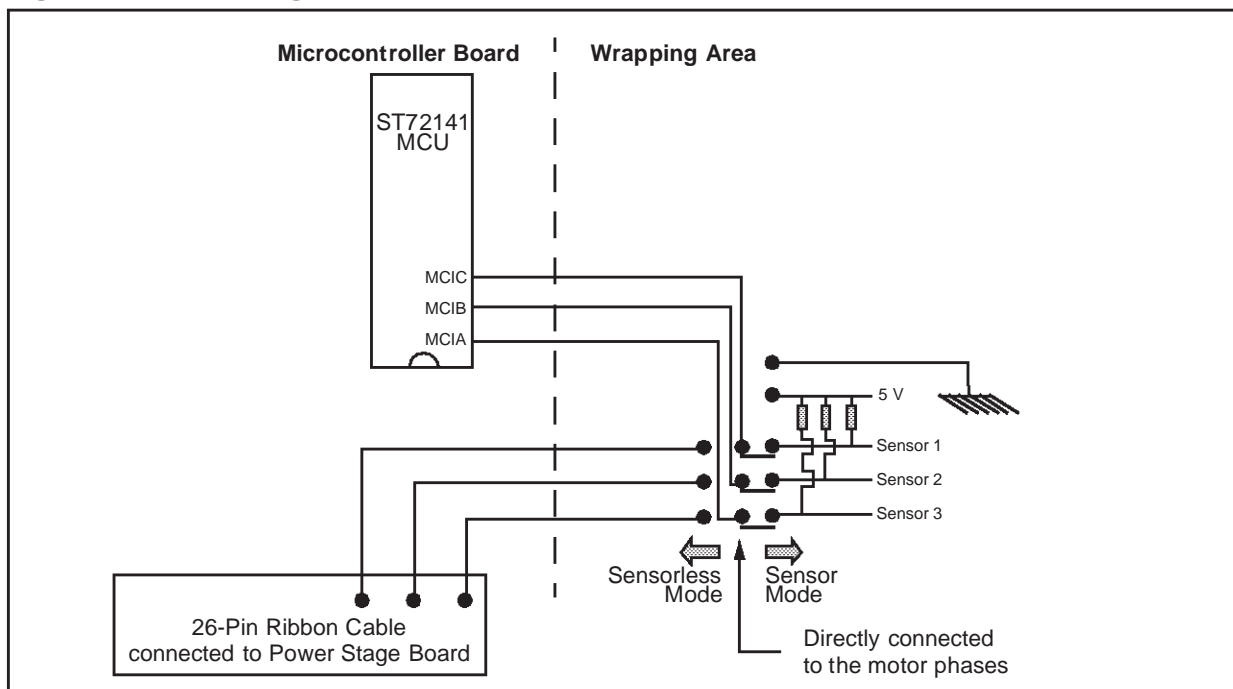
defined and the commutation tables written, you must not change the motor phase connections on the power stage board.

5.2 HARDWARE MODIFICATIONS

As explained in Section 1, the sensors must be connected to comparators MCIA, MCIB and MCIC. (The connection configuration is defined by the user.) If Sensorless mode is not required, the connection between the motor phases and comparators MCIA, MCIB and MCIC can be disconnected from the 26-wire ribbon cable and directly connected to the sensors via the wrapping area provided on the microcontroller board (don't forget the 4.7 kΩ pull-up resistors). In this case, the 5 V connection on the microcontroller board is used for the sensor power supply.

To maintain the Sensorless mode option on the board, a configuration that uses jumpers to switch between Sensor and Sensorless modes can be implemented. This configuration is shown in Figure 4. The first step is to disconnect the 3 wires coming from the 26-pin ribbon cable leading to the comparators in order to add the option of Sensor mode using the jumpers.

Figure 4. Connecting Sensors to the Microcontroller Board



As in the first case, the wires coming from the motor phases to the 3 comparators via the 26-pin ribbon cable must be disconnected. These wires are then connected to the wrapping area. Jumpers are used on the wrapping area to select either Sensor or Sensorless mode as shown above. The sensors are connected to these jumpers as well as to the 5V connection of the mi-

crocontroller board for the sensor power supply. To select either Sensor or Sensorless mode, simply change the jumpers.

The switches will behave better if pins 6 and 7 of each driver are soldered together on the power stage board.

5.3 SOFTWARE MODIFICATIONS

Only a few software modifications are needed to switch into Sensor mode:

- the constant file declaration must be modified in order to update the phase commutation tables for the motor,
- the main file must be modified at initialisation to validate Sensor mode in MCRA register,
- the position_detection routine must be modified at start phase of the motor in order to output the correct first step configuration according to the sensors state.

Once the transition table and sensor connections are determined, you can write your phase commutation tables. The phase commutation tables are already written in the software, but contain the default values that correspond to the Ametek motor provided with the kit. Only few bits have to be changed in the default phase commutation tables.

5.3.1 Modifications in the Constant Declaration File (.def source file)

In the Kanda kit, the constant declaration file cannot be modified directly. Only the initialisation, main and interrupts source files can be modified. This is a security feature because most of the constant values are declared and set in the ROM, except for the parameters modified in the previous stages of the kit and saved by the user. Therefore, a specific procedure is required in order to modify the constant file declaration. This procedure is described below.

For example, if the selected driving mode is Voltage Mode Closed Loop, the constant declaration file is the .def file named St7141vc.def. (This file be found in the directory of your computer where the kit has been installed). To modify this file, carry out the following steps:

- Make a backup copy of the St7141vc.def file.
- Open the file using a suitable text editor such as Notepad.
- Enter the required modifications (in the phase commutation tables).
- Save the modifications in the ST7141vc.def file.

Now, start the Stage 3 software, create a new project and select a suitable Voltage Mode Closed Loop .mtc file and go through the Application builder as normal, setting the peripherals as required. You will now find that the Stage 3 software is using your new St7141vc.def file.

If you wish to go back to the original file, delete the St7141.def file and rename your original backup copy as St7141vc.def.

The same procedure is used for each of the motor driving modes.

Note: The motor performances are improved in Sensor mode when the delay coefficient parameter is set

to 0/32 because sensor signals can frequently be assimilated to commutation signals.

What are exactly the changes required in the tables to update them to your motor?

Depending on the selected motor driving mode (current or voltage), there will be different tables in the constant file. **Only the following tables must be modified:**

- tables `phreg_SR` and `crbreg_SR` in Current mode,
- tables `phreg_SR` and `volt_crbreg_SR` in Voltage mode.

In table `phreg_SR`, only bits 6 and 7 of each line need to be modified. These bits will validate the correct comparator depending on the step. (Please refer to the ST72141 Datasheet for the MPHST register configuration (bits 7:6) to know the correct code for each comparator). The other bits do not require any modifications as they comply with the hardware configuration of the power stage board included with the kit.

In tables `crbreg_SR` or `volt_crbreg_SR`, only bit 5 of each line needs to be modified. This bit is the CPB bit of the MCRB register. Set this bit if the ST72141 must wait for a rising edge of the sensor on the step or reset it if the ST72141 has to wait for a falling edge of the sensor on the step.

5.3.2 Modifications in the Main File (.stm source file)

The main file can be directly modified in Stage 3 of the Kanda kit. The only modification required in the main file is to set bit 5 of the MCRA register during the initialisation phase. More precisely, in the `.volt_register` label and in the `.voltage` public label for Voltage Mode Open or Closed Loop driving modes and in the `.current_register` label and in the `.current` public label for Current Mode Open and Closed Loop driving modes. Set bit 5 of the MCRA register in the main file and save the file.

The software is now configured for Sensor mode and you can program a device.

Note: Setting bit 5 in the MCRA register will configure your software only for Sensor mode. Once your device is programmed, Sensorless mode will no longer be available (except if you return to Stage 3, reset bit 5 of the MCRA register and program another device).

Another solution is to configure a pin as a floating input. For example, pin PA3 or an other pin in the application builder may be used. If a switch is connected to this pin on the wrapping area, you can externally select Sensor mode or Sensorless mode. In this case, only the corresponding 2 lines of code for the selection of the same labels as before must be written. This will set or reset bit 5 of the MCRA register depending on the pin state. This solution avoids having to reprogram a device when you want to change the mode (Sensor or Sensorless).

5.3.3 Modifications to the Position_detection Sub-routine (.sts source file)

In the `position_detection` subroutine, the software checks the state of the comparator entry associated to comparators MCIA, MCIB and MCIC. Depending on the state of the pins, a different value is entered in a counter. For example when detecting the rotor position, we can see that if comparator MCIA is set to 1, the “count” parameter is incremented. In this case, if comparator MCIB is set to 1, the following instruction is executed:

```
LD A, count
or A, #2
LD count, A
```

to add 2 to the value of the counter. The same procedure applies if comparator MCIC is detected and the value associated is 4. This procedure gives a different set of values for the counter depending on the state of comparators MCIA, MCIB and MCIC. The values are then associated to the step configuration according to the Sensor state and transition table obtained with the sensors waveform (Table 1). This is shown in the code given in Section 2.2 on page 8.

For example, if the value of the “count” parameter is 7. This value corresponds to the detected MCIA, MCIB and MCIC configurations (1+2+4). According to the Sensor State and Transition table (Table 1), this configuration corresponds to step T3-T6 which is the third step in the 6-Step principle. As the pointer address starts at 0 in the table, in order to reach the third step, a value of 2 must be entered in the pointer (phasesstep). This is shown in the code example in Section 2.2 on page 8. Using the Sensor State and Transition table, the position_detection subroutine can easily be updated for your motor.

Note: If you want to use the Ametek motor that is provided with the kit in Sensor mode, the only software modification needed is the one in the main file (.stm source file) as shown in Section 5.3.2. Then the hardware modifications on the microcontroller board are the same. A reference manual for the Ametek motor is included on the CD-ROM. The software for the commutation tables for Sensor mode for this motor is based on the fact that:

- Sensor 3 is connected to pin MCIA,
- Sensor 2 is connected to pin MCIB,
- Sensor 1 is connected to pin MCIC,
- Motor phase A is connected to phase A entry of the power stage board,
- Motor phase B is connected to phase B entry of the power stage board,
- Motor phase C is connected to phase C entry of the power stage board.

Be careful when configuring your motor connections as the phase connections on the power stage board are not important in Sensorless mode. In certain kits, the motor phases may not be connected on the power stage board as described above.

6 CONCLUSION

This document shows how a ST72141 microcontroller is designed for Sensorless and Sensor Control of a BLDC motor. Sensor control of a motor is validated with this device and fully operational.

A complete software example for Voltage Mode Sensor control is attached. This software has been developed with Version 2 of the Kanda Motor Control Kit and the Ametek motor provided with the kit. Please note that at first the Kanda kit was not designed to support Sensor mode. The Kanda kit was first designed as a development kit for Sensorless driving mode. This document indicates how to modify the kit in order to work in Sensor mode. This document only

Using the ST72141 Motor Control MCU in Sensor Mode

contains the modifications required to operate in Sensor driving mode. The constant declaration file should not be modified as it complies with the generic software and hardware power stage configurations. This is why only the bits indicated in this document should be modified.

This software is for guidance only. STMicroelectronics shall not be held liable for any direct, indirect or consequential damages with respect to any claims arising from use of this software. Kanda does not guarantee that the boards (both power stage and microcontroller boards) will work correctly in Sensor mode.

Using the ST72141 Motor Control MCU in Sensor Mode

"THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNEXION WITH THEIR PRODUCTS."

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

The ST logo is a registered trademark of STMicroelectronics

©2001 STMicroelectronics - All Rights Reserved.

Purchase of I²C Components by STMicroelectronics conveys a license under the Philips I²C Patent. Rights to use these components in an I²C system is granted provided that the system conforms to the I²C Standard Specification as defined by Philips.

STMicroelectronics Group of Companies

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain
Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>