AN1276
APPLICATION NOTE

BLDC MOTOR START ROUTINE FOR THE ST72141 MICROCONTROLLER

by Microcontroller Division Applications

INTRODUCTION

The ST72141 microcontroller has been designed by STMicroelectronics to control BLDC motors in sensor and sensorless modes with a patented method for detecting the back-electromotive force zero-crossing event (see Application Note AN1130). The purpose of this document is to describe the motor start-up routine for the ST72141 microcontroller.

The start-up routine for a BLDC motor using the ST72141 microcontroller in sensorless mode consists of several phases. The first phase is the rotor prepositioning phase. In sensorless mode, we do not know the position of the rotor before starting the motor, that’s why we need to have a prepositioning phase. Then, a starting ramp table is applied to the motor in order to accelerate it and to be able to detect the information needed to switch to auto-commutated mode, this mode is called synchronous mode (see Application Note AN1083). During these different phases, several parameters of the generic software (supplied with AN1083) have to be taken into account. These parameters will need to be modified, depending on the motor behaviour. In this application note, we will explain the role of each of these parameters and the steps you need to follow to define the correct start-up parameters for your own motor.

Related Documentation

AN1082: Description of the ST72141 Motor Control Peripheral

AN1083: ST72141 BLDC Motor Control Software and Flowchart Examples

AN1129: PWM Management for Permanent Magnet DC Motor Drives using the ST72141

AN1130: Brushless DC Motor Drive with ST72141

Note: Contact STMicroelectronics for detailed technical information on startup algorithms for compressors (most leading manufacturers supported).

Table of Contents

INTRODUCTION 1

1 PREPOSITIONING PHASE 3

1.1 CONFIGURING THE PREPOSITIONING STEP 3

1.2 PREPOSITIONING CURRENT RAMP 5

2 STARTING RAMP TABLE 7

2.1 STARTING IN THE RIGHT DIRECTION 7

2.2 THE CURRENT LEVEL DURING THE STARTING RAMP 9

2.3 START SEQUENCE PARAMETERS 10

2.3.1 Forced synchronous mode 10
2.3.2 Consecutive zero-crossing events 11
2.3.3 Software demagnetization 11

2.4 APPLYING DECREASING STEP TIMES 12

2.5 STARTING RAMP TABLE PARAMETER GUIDELINES 14

3 SWITCHING TO AUTO-COMMUTATED MODE 15

3.1 THE LAST STEP IN SYNCHRONOUS MODE 15

3.2 THE FIRST STEP IN AUTO-COMMUTATED MODE 16

4 CONCLUSION: HOW TO PARAMETER YOUR MOTOR FOR THE START SEQUENCE 17

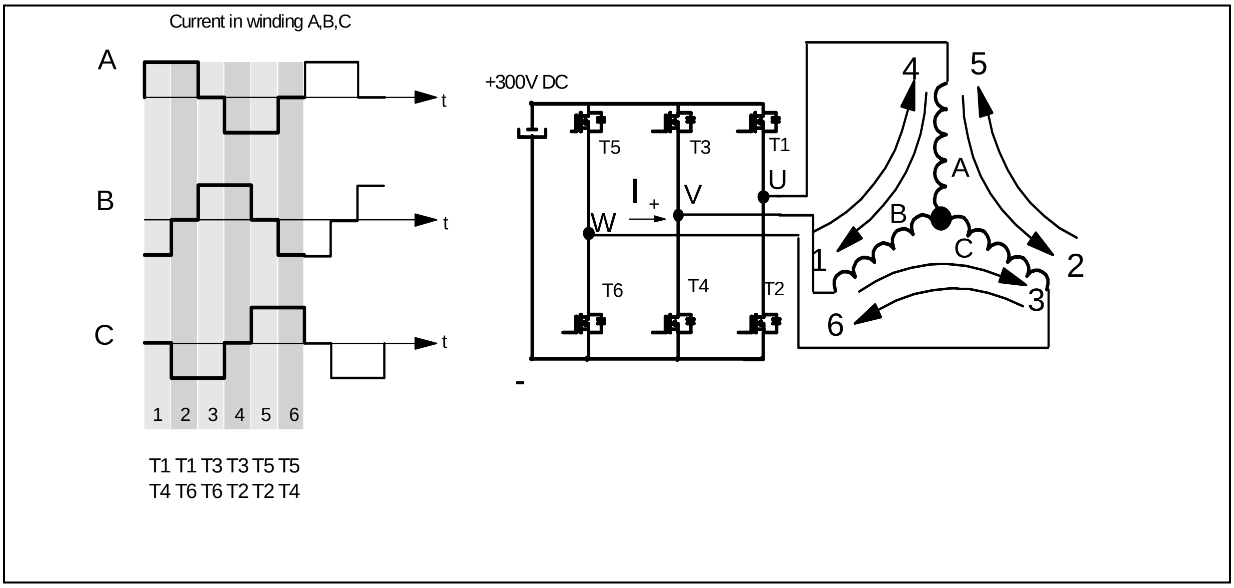
1 PREPOSITIONING PHASE

In sensorless mode, the position of the rotor before starting the motor is not known. A prepositioning phase is needed in this case either to determine the position of the rotor or to place it in an known position. With the ST72141, we use the prepositioning phase to place the rotor in an known location. This prepositioning phase is done entirely by software.

1.1 CONFIGURING THE PREPOSITIONING STEP

In the ST72141, the motor control is based on a six-step drive with a standard triple Half bridge (see Application Note AN1130). Figure 1 shows the six-step drive.

Figure 1. Six-Step drive



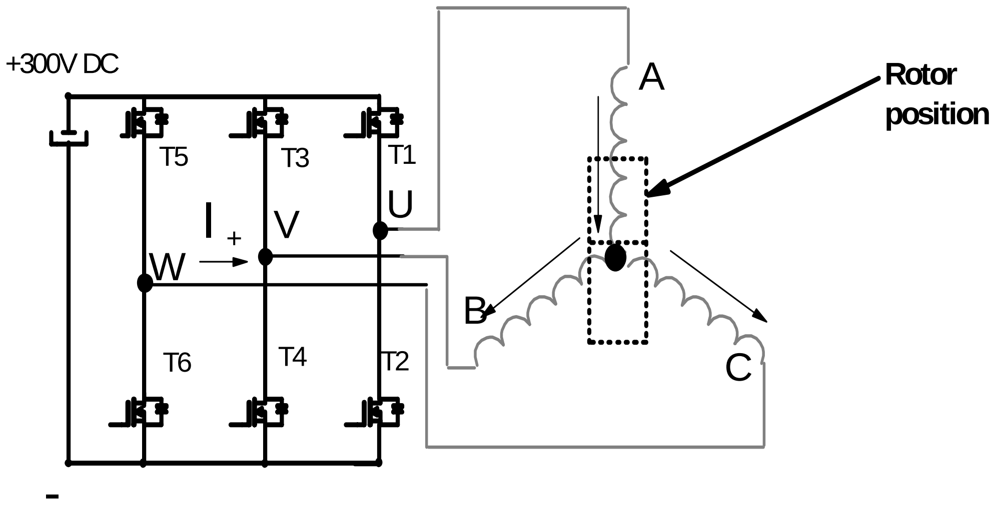
We see in Figure 1 that the six-step drive consists of a sequence where each winding is not energized at 2 times in the sequence. When a winding is not energized, it allows us to read the information needed (see Application Note AN1130). The step configuration sequence is (T1,T4), (T1,T6), (T3,T6), (T3,T2), (T5,T2), (T5,T4).

During the prepositioning phase, we want to put the rotor in a specific, known position. So, throughout this phase, we will keep a constant current direction in the windings to make the rotor move to the desired position.

In order to reduce the oscillations of the rotor when approaching its starting position, we apply current to all the windings. This means that all the windings will be energized during the prepositioning phase. This will be an intermediary position between two of the six step positions.

During the prepositioning phase, the T1, T4 and T6 switches are switched on. This means that when we will apply a current in this configuration, it will be positive in phase A and negative in phases B and C with half of the phase A value. This will bring the rotor to the intermediary position shown in Figure 2.

Figure 2. Rotor Start position

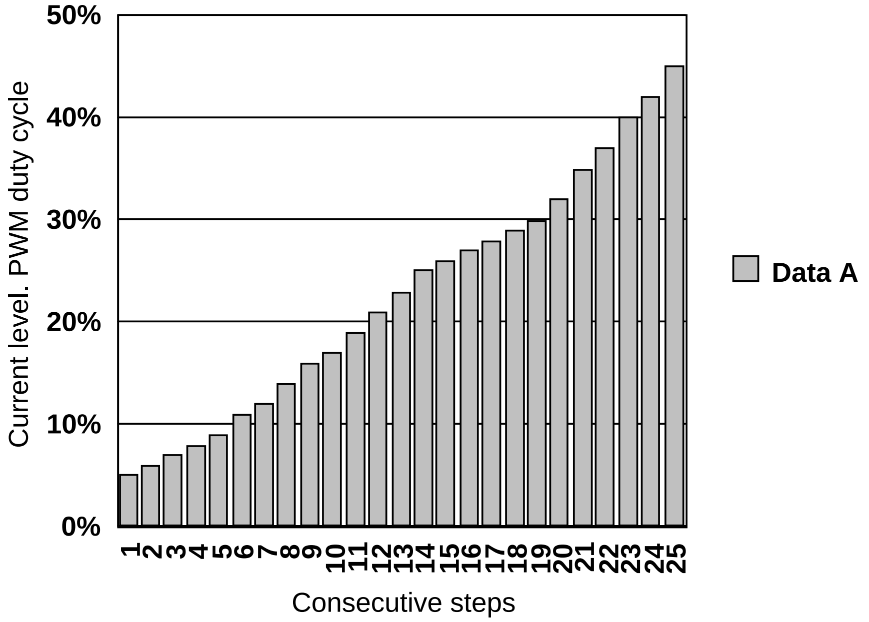


So, switching on T1, T4 and T6 will bring the rotor to the position shown in Figure 2 as soon as a current is applied. Since this position is known, you can then start the motor in the right direction with the correct step configuration for the starting ramp table (see Section 2).

1.2 PREPOSITIONING CURRENT RAMP

In order to bring the motor from its initial position (unknown by the user) to the prepositioning position, we will apply a current associated with the prepositioning step configuration (T1,T4,T6). Instead of applying a constant current value directly and waiting until the rotor arrives at the start position, we apply a progressive current level for a long enough time to bring the rotor in the desired position. Depending on its inertia, the rotor needs a certain level of current to move. Applying a strong current level directly to the rotor will make it move too quickly and this will make it oscillate around its final position. That’s why it is better to apply a progressive current level using a prepositioning current ramp. This ramp is made up of a certain number of steps, each step has the same length (this will give the prepositioning phase length) and each step has a corresponding current level. Figure 3 shows a waveform example of a 25-step current ramp, each step is 30ms long. This means a prepositioning phase of 750ms (30*25).

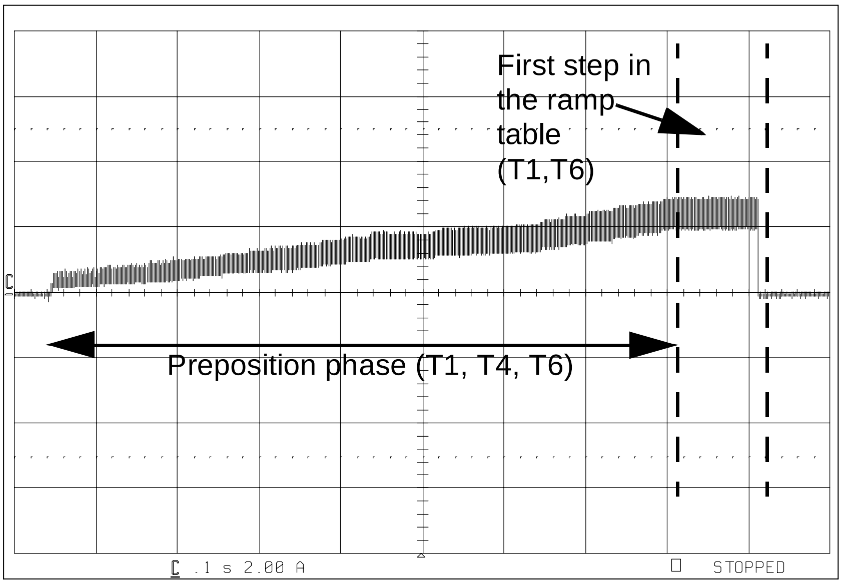
Figure 3. Prepositioning current ramp



The reference current or the reference voltage (depending if the motor is in current mode or in voltage mode) is given by the Timer A PWM signal (see Application Notes AN1130 and AN1082). This reference is represented by the Y axis in Figure 3, it is given by the duty cycle of the PWM signal. At a fixed PWM signal frequency, increasing the duty cycle will increase the reference. In this case, what will happen in the prepositioning phase, is that 25 times for a 30ms period, a different current level is applied with the same step configuration T1, T4 and T6. During this phase, T4 and T6 are switched on and the PWM signal is applied on T1 (see Application Notes AN1130 and AN1082 for the PWM signal application). The combined action of a constant step configuration and a progressive current ramp will bring the rotor into an intermediary position, the start position (Figure 2). This position is known by the user and the start sequence of the motor can begin.

Depending on the motor, a longer or a shorter prepositioning phase is needed. This is produced by defining a smaller or greater number of steps in the current ramp or making each step longer or shorter. The current level at each step and the ramp waveform also need to be modulated according to the motor. A low level at the end of the ramp may not be sufficient to make the rotor move and if the level is too high, it may cause the rotor to oscillate around the start position. Some motors will need a ramp with an exponential waveform, others a linear waveform. Several tests are needed to determine the best ramp for the motor, they consist of applying different ramps and examining the rotor behaviour. Figure 4 shows what is seen with a current probe on phase A of the motor during the prepositioning phase when applying the ramp in Figure 3.

Figure 4. Current in phase A during the preposition phase



We see in this figure that the current is applied progressively to the motor. The progression is linear. At the end of the prepositioning ramp, there is a current of 2 Amps in the motor.

2 STARTING RAMP TABLE

Once the rotor is in its start position, we apply a starting ramp table to the motor. The purpose of this ramp table is to accelerate the motor in order to detect the back-EMF zero-crossing information as soon as possible to in order to be able to switch to auto-commutated mode (see Application Note AN1130). This ramp table is made up of consecutively decreasing step times that are applied to the motor. Several conditions, set by the user and needed by the microcontroller, have to be met to allow the motor to enter auto-commutated mode. The purpose of this section is to list the different parameters corresponding to the different conditions that apply in the starting phase of a BLDC motor controlled with the ST72141 and to explain their influence on the motor behaviour.

2.1 STARTING IN THE RIGHT DIRECTION

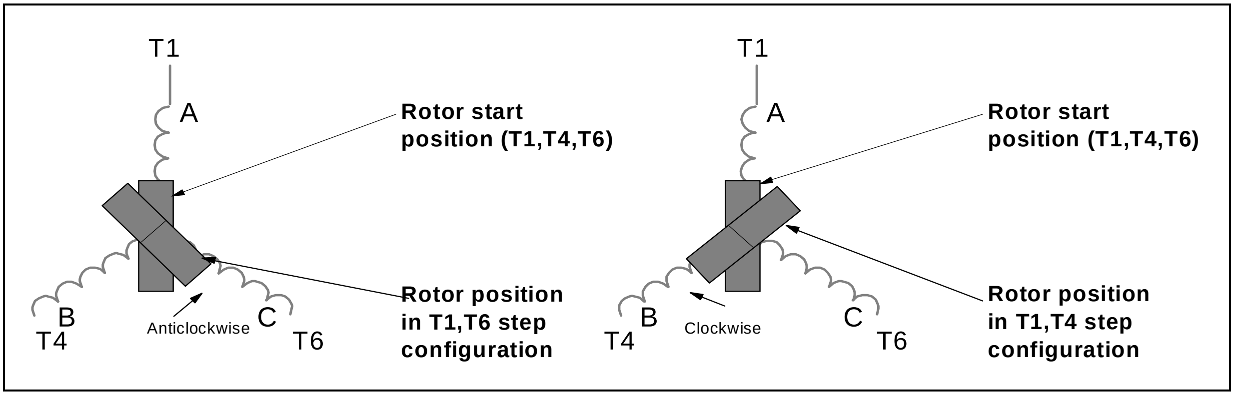
The six-step drive consists (as shown in Figure 1) of a succession of six different step configuration. In Figure 1, the step configuration sequence is:

step 1: switches T1 and T4
step 2: switches T1 and T6
step 3: switches T3 and T6
step 4: switches T3 and T2
step 5: switches T5 and T2
step 6: switches T5 and T4

Then the process goes back to step 1 etc....

This step configuration sequence will make, in our case, the rotor turn in the anticlockwise direction after the prepositioning phase. Figure 5 shows two different rotor positions corresponding to two different step configurations starting from the prepositioning position.

Figure 5. Two different rotor positions corresponding to two step configurations



We can see in Section Figure 5 that starting from the initial position with the T1, T4 step configuration makes the rotor turn clockwise. With the step configuration sequence defined in the software and listed above, our motor is intended to turn anticlockwise in the application so the first time we need to begin the sequence with the second step configuration which is T1, T6.

This step configuration, as shown in Figure 3, makes the motor turn anticlockwise from the initial position.

If the motor had to turn clockwise in our application, the first step configuration would have been T1,T4 in the starting ramp and the step configuration sequence in the six steps principle would have been inverted.

You need to modify this parameter if the initial position of the motor is different from the example given above, and some thought is required to determine the correct first step configuration to apply after the prepositioning phase.

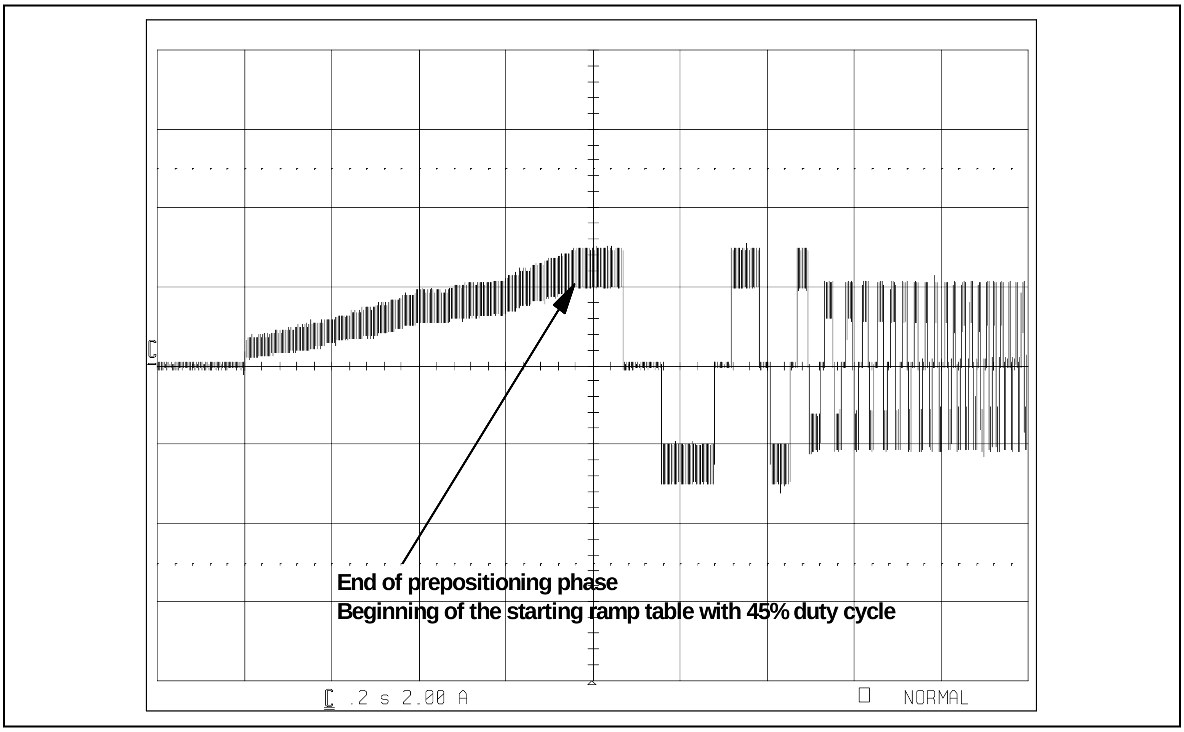
2.2 THE CURRENT LEVEL DURING THE STARTING RAMP

A specific reference current or voltage can be applied in the starting ramp table to accelerate the motor. This specific reference can be totally independent from the reference applied to the motor and chosen by the user when the motor is in auto-commutated mode. This current or voltage level during the starting ramp is chosen by the user and is set by software. Once again, as it is the case during the prepositioning phase and in auto-commutated mode, the reference is given by the Timer A with a PWM signal (combined with the internal clock signal in current mode). See Application Note AN1130.

At low speed, the back-EMF signal is weak and so harder to detect, applying a quite strong level of current during the starting ramp makes the back-EMF signal stronger and easier to detect at the zero-crossing event.

To avoid a surge of current at the beginning of the starting ramp after the prepositioning phase, it is better that the current level at the end of the prepositioning ramp is close to the current level during the starting ramp table. Figure 6 shows the prepositioning phase and the starting phase of the motor seen in the phase A with a current probe on an oscilloscope. This figure has been obtained with the prepositioning ramp shown in Figure 3 and a PWM duty cycle of 45% in current mode for the current level during the starting ramp.

Figure 6. Prepositioning phase & starting ramp table in current mode (45% duty cycle)



We can see in Figure 6 that the current level at the end of the prepositioning phase is the same as the one applied during the starting ramp table. This avoids a surge of current at the transition.

2.3 START SEQUENCE PARAMETERS

The starting ramp table consists of applying consecutively decreasing step times to the motor with a fixed current level for all the steps in order to accelerate the motor. The number of steps in the ramp is software programmable, in our case, we have chosen a 38-step ramp table. In all cases, if the motor arrives at the 38th step and has not yet switched to auto-commutated mode, it is better to stop the motor and to start it again. In normal use, the motor switches to auto-commutated mode after a few steps of the ramp table as soon as the detection of the back-EMF zero crossing event is enabled by the software.

The start sequence of the motor is based on different parameters that can be set by software. At first you can set the number of steps in the starting ramp table without enabling the detection of the back-EMF zero-crossing event (Z event). These steps (without Z detection enabled) are referred to as Forced Synchronous Mode, described in detail in Section 2.3.1. Then, you can set the number of consecutive Z events that have to be detected during the starting ramp table in order to switch to auto-commutated mode (see Section 2.3.2). Finally you can set the software demagnetization time during the starting ramp (see Section 2.3.3). The end of demagnetization event is very important in the motor start sequence. The purpose of the next two sections is to explain the role of these parameters on the start sequence of the motor.

2.3.1 Forced synchronous mode

During the first few steps at the beginning of the starting ramp table, the detection of the back-EMF zero-crossing event (Z) is disabled by software. These steps with Z detection disabled initially permit the motor to become stabilized, after these steps the motor will have accelerated enough to have a measurable back-EMF signal and zero-crossing event. Normally, the number of steps with Z detection disabled is equal to 1 mechanical cycle of the motor = p electrical cycles with p = number of motor pole pairs and 1 electrical cycle = 6 steps. It is not mandatory to disable Z detection for the first few steps. Z detection can also be enabled immediately at the beginning of the starting ramp table but generally you will see that the conditions for switching to auto-commutated mode are true only after a few steps. So, waiting a few steps before enabling Z detection will avoid a parasitic detection.

If you choose to wait one mechanical cycle before enabling Z detection, you have to pay attention to the fact that, depending on the number of motor pole pairs, this number of steps without Z detection can be quite high, but it must not be too near to the total number of steps in the starting ramp table. For example if you have a 10-pole motor, the number of steps without Z detection will be 5*6=30 and if the whole starting ramp is made up of 38 steps, 30 is too high, it has to be reduced. Otherwise, if Z detection is enabled only after 30 steps, you risk arriving at the end of the starting ramp table without detecting enough Z events. The motor won’t switch to auto-commutated mode and will be stopped. This depends of course on the number of consecutive Z events you want to detect before switching to auto-commutated mode.

On the other hand, if the number of steps without Z detection is too low, the motor will not be stable enough before the first Z event detection.

2.3.2 Consecutive zero-crossing events

After Forced Synchronous Mode, zero-crossing event (Z) detection is enabled. You can set the number of **consecutive** zero-crossing events to be detected by the microcontroller before allowing the motor to switch to auto-commutated mode. This number is a very important parameter. If this number is too high like 4 or 5 consecutive zero-crossing events, the risk is that the motor will finish the ramp table after the microcontroller has detected 2 or 3 consecutive Z events but without reaching the target number of Z events needed to switch to auto-commutated mode and the motor will be stopped at the end of the starting ramp. On the other hand, if this number is equal to 1, the risk is that one event will be detected, then software enables auto-commutated mode but the motor is not stabilized enough and in the first step in auto-commutated mode, there will be no Z event and the motor will stop. Two or three consecutive Z events are enough to allow the motor to switch to auto-commutated mode in a stable state.

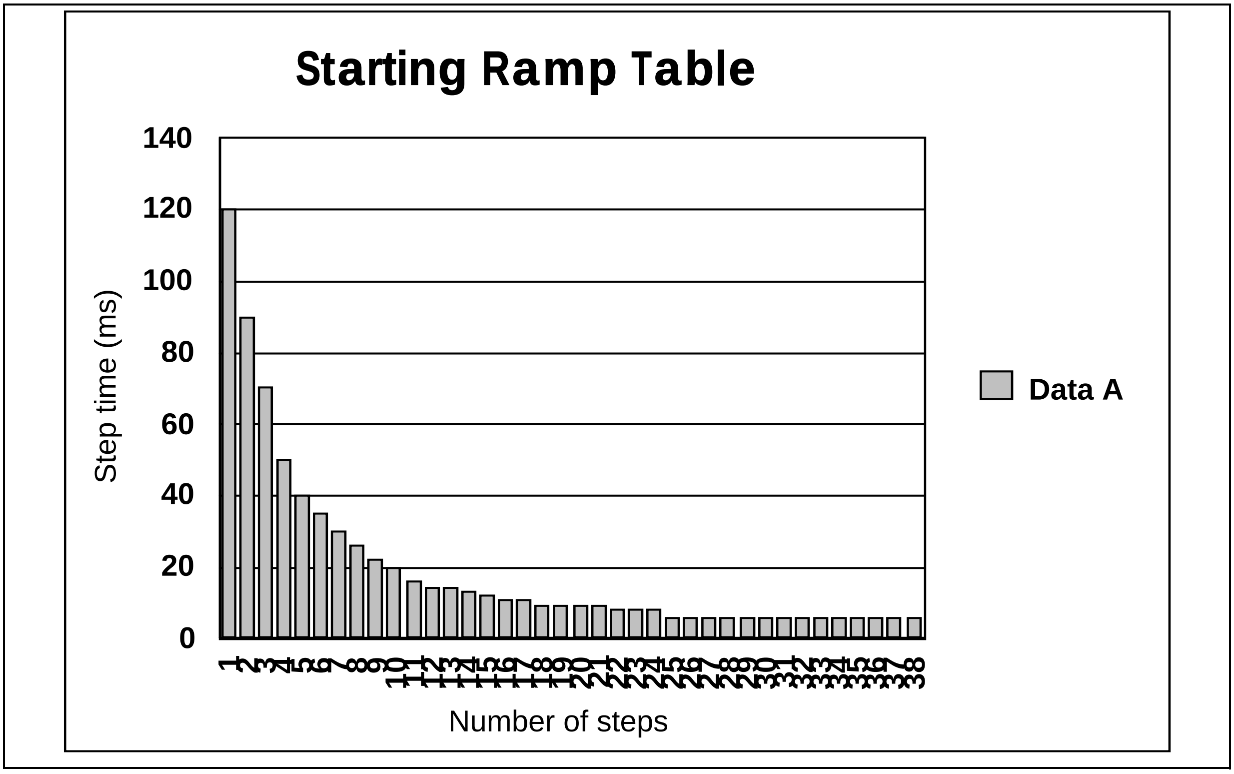
2.3.3 Software demagnetization

During the start ramp, hardware demagnetization and software demagnetization are performed on alternate steps. This means that in the steps where we wait for a falling edge of the phase voltage signal for the end of demagnetization event detection there will be a software demagnetization and in the steps where the end of demagnetization event will be a rising edge of the phase voltage signal, there will be a hardware demagnetization. Each software demagnetization time is equal to 1.25 times the hardware demagnetization time in the previous step. This means that the software demagnetization time is directly proportional to the hardware demagnetization time. Avoid setting the factor too high because if the software demagnetization time is too long, the risk is that the microcontroller won’t see the zero-crossing event, especially during the start sequence where the signal is quite weak. On other hand, this factor has to be greater than 1 to be sure that the end of demagnetization event really happened. Even if we choose a software demagnetization, the time set for that event has to be long enough to be appear like the corresponding hardware demagnetization event.

2.4 APPLYING DECREASING STEP TIMES

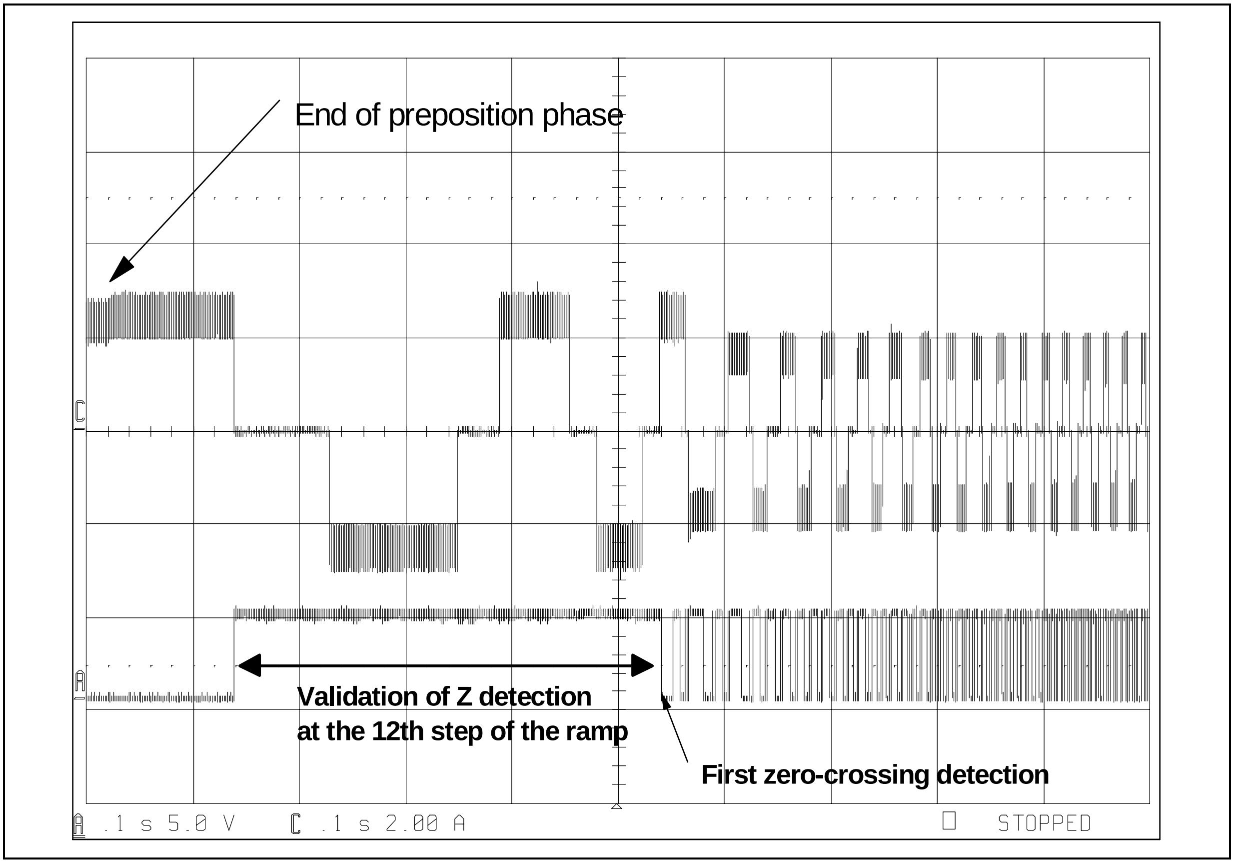
The difficulty in defining the right starting ramp table for a motor lies in the fact that the sequence of decreasing step times must accelerate the motor enough to be able to detect a zero-crossing event after Forced Synchronous Mode and still obtain the target number of consecutive events to switch to auto-commutated mode. In the same way as for the prepositioning ramp, the waveform of the starting ramp table will depend on the rotor inertia and the motor load. You can determine the best ramp table for a given motor by performing several tests. Figure 7 shows a waveform example for a starting ramp table with 38 steps. 38 steps is just an example, we have chosen this number for our software but you can set a higher or a lower number of steps.

Figure 7. Example of a 38-step starting ramp table



This starting ramp has an exponential acceleration. Some motors will need a linear acceleration. The best way to find the right ramp table is to put a flag in the software on the zero-crossing event detection and then to test several ramps and see the result on an oscilloscope with the current signal on a phase to see the ramp table and a probe on the flag for zero-crossing detection. When a ramp makes the rotor move and accelerates it enough to detect several zero-crossing events and at least 2 consecutive events before the end of the ramp, it is a good ramp for that motor. Figure 8 shows a view using an oscilloscope with a current probe on the phase A and another probe on the flag for zero-crossing detection during a start sequence.

Figure 8. Starting ramp table and zero-crossing event overview



We see 3 different phases in these waveforms in Figure 8.

First phase: The end of the prepositioning phase

Second phase: Forced synchronous mode (no zero-crossing event detection)

Third phase: Zero-crossing detection before switching to auto-commutated mode

The flag for the zero-crossing event in the software is set on the commutation event and reset on the back-EMF zero-crossing event in this example.

2.5 STARTING RAMP TABLE PARAMETER GUIDELINES

Three main parameters are important during the start sequence after the prepositioning phase.

The number of steps in forced synchronous mode (no zero-crossing event detection)
The number of consecutive zero-crossing events before switching to auto-commutated mode
The starting ramp table waveform (application of decreasing step times)

Each of these parameters has an influence on the motor behaviour during the start sequence and is software programmable.

The number of steps in Forced Synchronous Mode needs to be large enough to obtain a well-stabilized motor. Too high a number, too close to the total number of starting ramp steps number will result in finishing the ramp table without detecting enough consecutive zero-crossing events to switch to auto-commutated mode.

The number of consecutive zero-crossing events is also very important. Too high a number has the same effect as too high a number of steps in Forced Synchronous Mode, meaning that we risk arriving at the end of the starting ramp table without detecting enough zero-crossing events. Too low a number will cause the motor to switch too quickly in auto-commutated mode with the risk that the motor stops because it is not stabilised enough.

The starting ramp table waveform has to make the rotor move and accelerate it enough to allow the detection of the target number of consecutive zero-crossing events to switch to auto-commutated mode.

Finding the best ramp table for the motor is achieved by performing several tests with an oscilloscope to see the waveform and the zero-crossing detection.

3 SWITCHING TO AUTO-COMMUTATED MODE

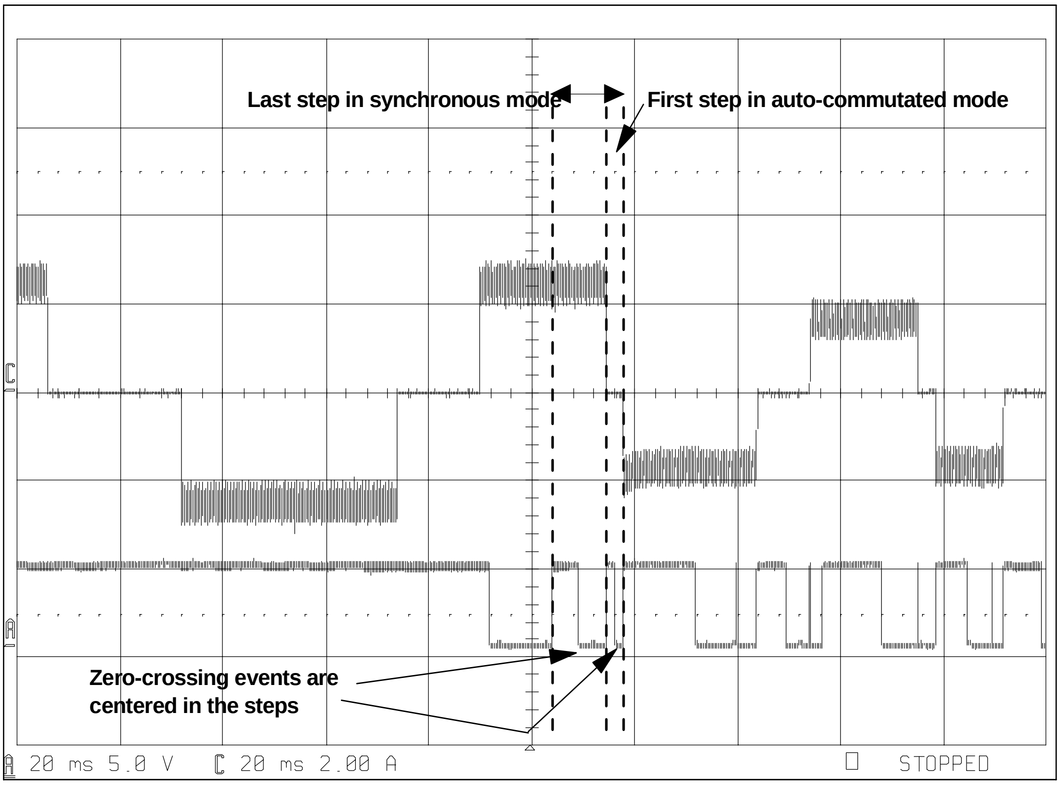
At the end of the start sequence, once the microcontroller has detected the target number of zero-crossing events, it switches to auto-commutated mode. This switchover is very important for the motor behaviour and once again, you have to pay a lot of attention to programming the parameters. This is described in detail in the two following sections.

3.1 THE LAST STEP IN SYNCHRONOUS MODE

For example, if you have set a target of two consecutive zero-crossing events before switching to auto-commutated mode, when the microcontroller detects the second one, it will memorize that the motor is in its last step in synchronous mode (see Application Note AN1083). In the generic software (supplied with AN1083), the last zero-crossing event detected before switching to auto-commutated mode is centered in this step. You can place this event wherever you want in the step because in synchronous mode, the step time is imposed and not computed automatically like in auto-commutated mode (see Application Note AN1130).

Centering this last zero-crossing event on the corresponding step allows you to predict the place where the next event will be. The step after this last step in synchronous mode will be the first step in auto-commutated mode, we do not know where the zero-crossing event will take place on this step but we are sure that there will be one. The time of this first step in auto-commutated mode depends on when the zero-crossing event occurs because the delay will be automatically computed, so, we do not know the step time of the first step in auto-commutated mode. Centering the event on the last step in synchronous mode allows us to have a step time in this step closer to the next step in auto-commutated mode than if we had not centered the event. Figure 9 shows what we can see on an oscilloscope when the switch to auto-commutated mode occurs.

Figure 9. Switching to Auto-commutated mode



We can see in Figure 9 that the last zero-crossing event is centered in the last step before switching to auto-commutated mode.

3.2 THE FIRST STEP IN AUTO-COMMUTATED MODE

During the first step in auto-commutated mode, one parameter is very important, the delay factor (see Application Notes AN1130 and AN1082). This parameter is the one used by the microcontroller to automatically compute the delay between the actual zero-crossing event on this step and the next commutation event. The operation to find the real delay done by the microcontroller is: 8-bit timer value* delay factor/32 (see Application Notes AN1130 and AN1082). In auto-commutated mode, the timer contains the current step time value (this means the time between 2 zero-crossing events) but in the first step in auto-commutated mode, the timer contains the value between the last commutation event and the actual zero-crossing event on the step. This because in synchronous mode, the timer is reset on a commutation event and in auto-commutated mode, the timer is reset on a zero-crossing event. So, because of this particularity, the delay factor in the first step in auto-commutated mode can be different from the factor used later. This first factor value has to approximately center the zero-crossing event. The unit of the delay factor is 1/32, so a delay factor of 32 will center the zero-crossing event in the first step in auto-commutated mode. A delay factor of 8 will put the zero-crossing event at 3/4 of the step.

4 CONCLUSION: HOW TO PARAMETER YOUR MOTOR FOR THE START SEQUENCE

The first parameter you have to set for the start sequence is the prepositioning ramp. This ramp has to be tested in order to avoid the rotor oscillating when it is at the start position with the initial step configuration. You have to apply enough current to make the rotor move but a too high level of current during the whole ramp will induce oscillations.

Then, when you have fixed the prepositioning current ramp, the second parameter you have to set is the starting ramp table with the level of current during this ramp table. The starting ramp table has to accelerate the rotor enough to be able to detect the zero-crossing event of the back-EMF signal. The level of current during this ramp will help to accelerate the rotor. You can test several ramps with different current levels and connecting an oscilloscope with a current probe on one of the motor phases will help you to see the behaviour of the motor.

Afterwards you set the conditions for the motor to switch to auto-commutated mode, this means setting the target number of consecutive zero-crossing events. To set this number, it is better to set a flag on a zero-crossing interrupt in the software and use an oscilloscope to see how the detection of zero-crossing events appears throughout the whole ramp. To do this, it is better at first to enable the detection of the zero-crossing events immediately at the beginning of the starting ramp. This will allow you to see when there are 2 or 3 or more zero-crossing events in the ramp and this will help you to set the number of steps without Z detection and the target number of consecutive zero-crossing events before switching to auto-commutated mode.

Switching to auto-commutated mode:

Once the starting ramp and the conditions for switching to auto-commutated mode are fixed, there are two parameters that you have to set for auto-commutated mode. The start delay factor in the first step in auto-commutated mode (this will determine the position of the zero-crossing event detected in the first step in auto-commutated mode) and the delay factor for auto-commutated mode after the first step. Be aware that they can have different values.

To set all the preceding parameters for your own motor will take some time. Each of them has a specific role and this means you will have to perform several tests to determine the best values for all of them. You will need to use an oscilloscope to carry out these tests, it will help you understand the motor behaviour.

“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

©2000 STMicroelectronics - All Rights Reserved.

Purchase of I^{2}C Components by STMicroelectronics conveys a license under the Philips I^{2}C Patent. Rights to use these components in an I^{2}C system is granted provided that the system conforms to the I^{2}C 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