HT46R14A Single Phase AC Induction Motor Frequency Converter Application

D/N : HA0095E

Introductions

Initially the main reason for using frequency conversion technology was for speed control, however to consider the needs of power control, today’s frequency converters in addition to being used for speed control are also used for energy conservation purposes. Frequency conversion control technology is already being applied in many application areas, in home appliances such as fridges, washing machines, air-conditioners in addition to driving industrial motorised equipment etc. Frequency converters when used in electric motor control, are used to both control the frequency and voltage. The Holtek HT46R14A MCU device is used here in an example to show how frequency conversion is used for electrical motor control. An actual circuit is supplied for the users consultation.

Frequency Converter Principles

The normal household power supply or industrial power supply voltage and frequency is well defined and not open to change. The device, which takes this fixed voltage and frequency AC power and converts it into a variable voltage or variable frequency AC supply, is known as a Frequency Converter. In order to generate a variable voltage and frequency, the system must first rectify the AC input power source into DC after which the DC will be reconverted back into AC using the converter. The device, which converts the DC back into AC, and which can be controlled in both voltage and frequency, is known as the frequency converter. The frequency converter’s output is a simulated sine wave whose main purpose is to control the speed of an asynchronous induction motor.
Because of the characteristics of electric motors, most frequency converters must supply a Voltage/Frequency means of control. When the frequency is changed the voltage must change with it. When the frequency is reduced the voltage must reduce as well. If the frequency is reduced but the voltage does not change, the electric motor can easily overheat and burn out. This is especially true for very low frequencies. Therefore the frequency converter must be able to produce both voltage and frequency changes. In addition, most frequency converters also provide an acceleration and deceleration function, for use in emergency increase or decrease applications and for slow increase and decrease speed applications.

Most frequency converter's application area is for three phase AC induction motors. Fig. 1 shows a simple open-loop three phase output frequency converter block diagram. Most frequency converters supply single phase or three phase AC outputs. After bridge rectifying and filtering, the DC power is provided to the power drive circuits. The SMPS will convert the high voltage DC to the required isolated DC voltages and supply it to the protection circuits, drive converter interface circuits and the control circuits. The control circuits will provide the necessary control signals to drive the power circuits to control the motor's rotational speed and voltage. The frequency converter needs to provide protection circuitry for over-voltage, over-current, low-voltage, etc. which are used to protect the converter from adverse conditions. The driver converter frequency converter interface is an MCU control circuit and is an interface to the power drive circuit and provides voltage isolation. The usual way of providing isolation is with IC opto-couplers.
Example Circuit Functional Description

The example circuit has at its centre a Holtek HT46R14A MCU. Its main function is to provide driving signals for a single phase AC induction motor. The important circuits include high voltage circuits, low voltage circuits, power driving circuits, drive converter interface circuits, protection circuits, control circuits etc.

High Voltage Circuits
The main parts here are a bridge rectifier and filter, to power the main power high voltage DC circuits.

Low Voltage Circuits
The AC first passes through a transformer to reduce the voltage and then a bridge rectifier, filter and regulator to provide two voltage levels, 12V and 5V, which are provided to the driver interface circuits, protection circuits and the MCU controller circuits.

Power Driving Circuits
The Power Driving Circuits are composed of power switches which are used to drive the load. Such applications usually utilise IGBTs or Power Mosfets. This circuit uses four IRF840 Power Mosfet transistors, to form a bridge driver circuit, which is used to drive a single phase AC induction motor load.

Driver Converter Interface Circuit
The driver converter interface circuits sits between the MCU controller circuit and the power driver circuits. It provides voltage interface conversion, supplies Dead Time protection and Shut Down function protection etc. The present circuit uses two IR2109 devices as a driver converter interface.
Protection Circuits
In the present example the protection circuits are for over-current protection and will protect the system from overload situations. The designer may wish to add further protection such as circuits for over-voltage, low-voltage etc. This circuit’s output is connected to the SD pin of the IR2109 device and the MCU INT pin. Should a situation of over-current present itself, then the protection circuit’s output pin will go from high to low which will shut-down the IR2109 and generate an MCU interrupt. The MCU will then return to its reset condition and stop outputting information.

Controller Circuits
The controller circuit uses the Holtek HT46R14A MCU device’s PPG function as the central part of the system. The MCU controller circuit provides the following functions:

- Uses a VR for speed control, uses an ADC to monitor the desired speed
- Uses the PFD to control the 32kHz carrier wave output, uses the PPG to control the PWM duty cycle signal
- Controls the frequency output to have a range between 10Hz and 120Hz
- Provides a two stage V/F control for increased deceleration function
- Over-current stop motor protect function
- Display over-current and rotation status
- Start and stop switch functions

Software Description

**MCU Hardware Function Description**

The HT46R14A MCU has an operating frequency of 16MHz and a PFD output function connected to COMP1 and negative terminal C1VIN- and the INTOB pin to act as the PPG0 and PPG1 signal source. A VR is used to control the speed. INT0B is used as an over-current interrupt input source. When a situation of over-current exists, an interrupt will be generated which will stop the system from operating and illuminate an LED to indicate an over-current status has occurred. The system also provides a system running LED. A start and stop switch is provided to control the motor start and stop function.
MCU Software Function Description

- Timer 1 is the PFD clock source. The main function of the PFD is to generate a carrier wave frequency to supply a signal source for PPG0 and PPG1. The carrier wave in this example is setup using the program to be 32kHz. The designer can setup different values according to their requirements, by changing this value. When changing this value, the PPG0 and PPG1 preload value must also be changed.

- Timer 0 is used to control the output frequency which is used to control the rotational speed of the motor. Its prescaler and internal values are setup by the VR rotational position. After an ADC conversion the value is placed into the Timer 0 register. Timer 0 internal register value is:
  \[
  \text{Timer } 0 = \frac{16M}{25/128/2/ \text{VR actual rotational value}} = \frac{2500}{\text{VR}}
  \]
  "16M" is the system frequency
  "25" is the steps of the 180° degree sine wave
  "128" is the Timer 0 prescale value

- PPG0 and PPG1 are PWM outputs, which create sine wave positive and negative half period outputs. A PWM output has a fixed frequency but has a variable pulse width. The system has a sine pointer which reads a sine stable step by step. When the actual rotational frequency is greater than 60Hz, the PPG0 and PPG1 values will be the values read from the sine table. When the frequency is less than 60Hz, then the PPG0 and PPG1 values will be equal to the VR actual value \( \times \) sine table value \( \div 60 \). The sine table values are a 180° degree sine wave divided into 25 rotational sections.

- Rotational speed is controlled by the VR. After conversion by the A/D it will obtain a value of between 10 and 120, and after some manipulation will be placed into the Timer 0 register. This allows the rotational speed to have a range between 10Hz and 120Hz.

- The software outputs a frequency between 10Hz to 60Hz which is controlled by the V/F relationship. There are two settings for the V/F which is selected by jumper JP2, to select either the standard V/F or a V/F2 setting as shown in the diagram. If the jumper JP2 is shorted then the V/F2 setting is selected.

\[ \text{V/F Relationship} \]

\[ \begin{align*}
&\text{V} \\
&\text{f (Hz)} \\
&10 \\
&60 \\
&120
\end{align*} \]
• When changing the time of the acceleration and deceleration, the motor driver's startup acceleration rate can be used to obtain more gentle acceleration and deceleration. The AC motor driver will, according to the acceleration time, allow for different acceleration and deceleration curves.

• INT0B is used for the over-current interrupt input source. When an over-current situation occurs an interrupt will be generated and the software will stop the PPG0 and PPG1 outputs and return to the initial conditions to protect the system.
FlowChart: Time Division Multiplexing

a. Main Flowchart

1. Start
2. Initiate AD Interrupt
3. Initialize maincon
4. ADC Interrupt Finished?
   a. if YES then
      i. Main Flowchart
   b. if NO then
      i. YES
      ii. Keyboard
      iii. Display
5. Setup as Reset Procedure (Keyboard)
6. Setup as Reset Procedure (Display)
7. Setup as Reset Procedure (SPEED 1)
8. Setup as Reset Procedure (SPEED 2)
9. Setup as Reset Procedure (SPEED 3)
10. Setup as Reset Procedure (SPEED 4)
11. Keyboard
12. Display
13. Keyboard
a-1. Speed Conversion Flow (1)

1. Speed conversion procedure (1)

   AD data added 8 times?
   
   NO
   YES

   AD data / 8

   Add deceleration data \(-1\)?
   
   NO
   YES

   Key ON? conversion condition?
   
   NO
   YES

   VR actual rotational value = VR average?
   
   NO
   YES

   VR reduce actual rotational value by "1"

   VR actual rotational value = 0?
   
   NO
   YES

   VR increase actual rotational value by "1"

2. Setup next procedure speed convert (2) return to main program loop
a-2. Speed Conversion Flow (2)

Speed conversion flow (2)

On switch pressed? Rotation condition?

YES

NO

VR rotational value > rotational lowest limit

YES

NO

Already start rotation first data value conversion finished?

YES

Clear Sin Pointer read Sine Table value

NO

Clear Sin Pointer setup Timer1 off Except for key setup with initial value

VR rotational value < rotational lowest limit

YES

NO

Setup Timer0 ON setup Timer ON setup LED display

Setup reset procedure conversion (3) main program loop
a-3. Speed Conversion Flow (3)

1. AD conversion finished?
   - YES: Data calculation \( \frac{2500}{VR} \) rotational value
   - NO: VR rotational value > smallest rotational limit
2. VR rotational value > smallest rotational limit?
   - YES: Setup data conversion finished
   - NO: Setup reset procedure speed conversion (4) return to main loop
a-4. Speed Conversion Flow (4)

Speed Conversion Flow (4)

Data conversion finished?

YES

NO

VR rotational value < smallest rotational limit

YES

VR rotational value > longest rotational limit

NO

Data conversion value compensation load into Timer 0

Load smallest conversion limit into Timer 0

Load largest conversion limit into Timer 0

Setup next procedure display return to main loop
b. AD Interrupt Subroutine

ADC interrupt subroutine

set maincon.0

Read A/D converter value

reti

c. EXT INT0 Interrupt Subroutine

EXT INT0 subroutine

set pb.4 LED display over-current

Initialize system

reti
d. Timer0 Interrupt Subroutine

- Save main program register data
- Sine Pointer add one
- real Sine Table value
  - Clear Sine Pointer value
    - NO
      - VR rotational value >> 60
        - NO
          - V/F 1 PB.5=1 ?
            - NO (VF2)
              - VR rotational value +80H
                - Read conversion table value
                  - V/F conversion calculation management
                    - Data compensation load into PPG0 Timer PPG1 Timer
                      - Restore main program registers
                        - reti
Application Circuit Example
<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA3/PFD</td>
<td>Enable</td>
</tr>
<tr>
<td>PPG1</td>
<td>Enable</td>
</tr>
<tr>
<td>Pull-high PC0~PC3</td>
<td>Non-Pull-high</td>
</tr>
<tr>
<td>PFD Clock Source</td>
<td>from Timer 1</td>
</tr>
<tr>
<td>OSC</td>
<td>Crystal</td>
</tr>
<tr>
<td>XTAL OSC</td>
<td>&gt;10MHz Mode</td>
</tr>
<tr>
<td>WDT</td>
<td>Enable</td>
</tr>
<tr>
<td>WDT Time out select</td>
<td>WDT Clock source/2^16</td>
</tr>
<tr>
<td>WDT Clock Source</td>
<td>WDTOSC (12kHz)</td>
</tr>
<tr>
<td>CLRWDT</td>
<td>One clear instruction</td>
</tr>
<tr>
<td>LVR</td>
<td>Enable</td>
</tr>
<tr>
<td>PPG0 output</td>
<td>Active high</td>
</tr>
<tr>
<td>PPG0 timer</td>
<td>Synchronized with clock</td>
</tr>
<tr>
<td>PPG1 output</td>
<td>Active high</td>
</tr>
<tr>
<td>PPG1 timer</td>
<td>Synchronized with clock</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pull-high PA0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wake-up PA0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pull-high PB0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
PCB and Demo Board
Revision History

Revision: V1.10

Updated Date: 2007/07/16

Modified Contents: The original HT46R14 part number has been modified to HT46R14A.