Tác giả: TAHMID MAHBUB

Khi tạo ra sóng sin bằng kỹ thuật số, mình làm như vậy bằng cách sử dụng một số xung sóng vuông cố định. Con số này càng lớn thì sóng hình sin càng sạch.
Dạng sine y=sin(x) (x thay đổi giá trị từ 0-360 độ)

VD ta sử dụng 10 bước/bán kỳ. Một nửa bán kỳ sau mình sẽ đảo lại
Vì mình có 10 bước/bán kỳ, nên mình chia nửa dương của sóng sin thành 10 phần. Phần đầu tiên là 0-18 độ, xung tiếp theo ở 36 độ, sau đó là 54 độ, v.v.

Vì vậy, các giá trị mình cần là:

- 1. Sin(0) = 0
- 2. Sin(18)= 0.31
- 3. Sin(36)= 0.59
- 4. Sin(54) = 0.81
- 5. Sin(72)= 0.95
- 6. Sin(90) = 1
- 7. Sin(108) = 0.95
- 8. Sin(126) = 0.81
- 9. Sin(144) = 0.59
- 10. Sin(162) = 0.31
  Ta không lấy giá trị điện áp tại sin(180) vì giá trị này sẽ được gọi tiếp khi thực hiện sine(0)=0. Sau khi biết cách tìm các giá trị tại mỗi bước. Bây giờ, làm thế nào để thực hiện điều này bằng PWM?. 1 nghĩa là chu kỳ nhiệm vụ 100%, 0 nghĩa là 0% và 0,5 nghĩa là 50%. Vì vậy, từ bảng trên, mình biết rằng:
- 1. Sin(0) = 0%
- 2. Sin(18)= 31%
- 3. Sin(36)= 59%
- 4. Sin(54) = 81%
- 5. Sin(72)= 95%
- 6. Sin(90) = 100%
- 7. Sin(108) = 95%
- 8. Sin(126) = 81%
- Sin(144) = 59%
- 10. Sin(162) = 31%

Bây giờ, mình cần đặt các giá trị này vào một bảng sin thích hợp. Để làm được điều đó, mình cần biết giá trị của PR2. Khoảng thời gian PWM = [(PR2) 1] * 4 * TOSC * (TMR2 Giá trị đặt trước) mình biết, khoảng thời gian PWM, TOSC và giá trị đặt trước, vì vậy đặt giá trị này và sắp xếp lại cho mình, PR2 1 = Khoảng thời gian PWM / (4 * TOSC * {TMR2 Prescale Value})
Do đó, PR2 = [PWM Period / (4 * TOSC * [TMR2 Presca le Value})] – 1 * Công thức bắt nguồn từ đó được đưa ra trong biểu dữ liệu.
Bây giờ, mình cần quyết định xem mình sẽ thực hiện PWM (tần số sóng mang) ở tần số nào. Giả sử, mình có một bộ dao động 16MHz và mình quyết định sử dụng tần số sóng mang là 16kHz. Sau đó, nếu bạn đặt các giá trị vào công thức trên, mình có:
PR2 = [(1 / 16.000) / (4 * [1 / 16.000.000]. * 1] – 1 Cho mình, PR2 = 249

Vì vậy, mình gán 249 cho PR2. Thanh ghi chu kỳ nhiệm vụ là CCPR1L. Khi CCPR1L = 0, chu kỳ nhiệm vụ = 0, khi CCPR1L = (PR2 1), chu kỳ nhiệm vụ = 100%. Điều này là do (PR2 1) = Chu kỳ
Vì vậy, CCPR1L / (PR2 1) = Chu kỳ hoạt động Vì vậy, bảng:

- 1. Sin(0) = 0%
- 2. Sin(18) = 31%
- 3. Sin(36) = 59%
- 4. Sin(54) = 81%
- 5. Sin(72) = 95%
- 6. Sin(90) = 100%
- 7. Sin(108) = 95%
- 8. Sin(126) = 81%
- 9. Sin(144) = 59%
- 10. Sin(162) = 31%

Có thể được sắp xếp lại như sau:

- 1. Sin(0) = 0* 250
- 2. Sin(18) = 0.31 * 250
- 3. Sin(36) = 0.59 * 250
- 4. Sin(54) = 0.81 * 250
- 5. Sin(72) = 0.95 * 250
- 6. Sin(90) = 1 * 250
- 7. Sin(108) = 0.95 * 250
- 8. Sin(126) = 0.81 * 250
- 9. Sin(144) = 0.59 * 250
- 10. Sin(162) = 0.31 * 250

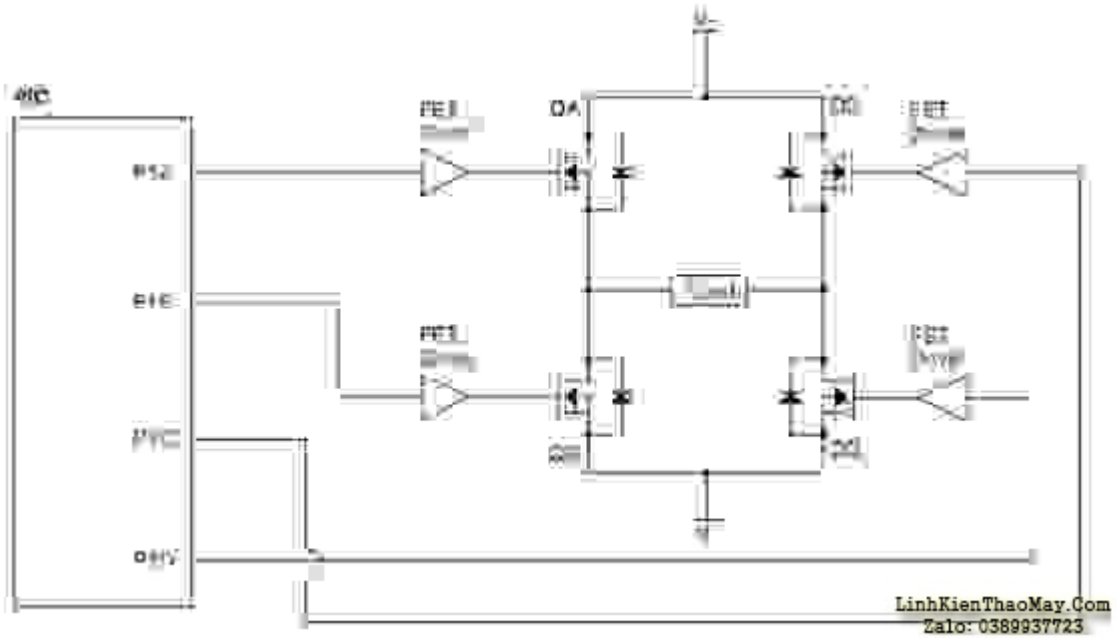
Ở đây mình nhân với 250 vì 250 là đỉnh của sóng sin [y = sin (x) * 250] với 250 là chu kỳ.
Vì vậy mình có:

- 1. Sin(0) = 0
- 2. Sin(18) = 77.5, take as 77
- 3. Sin(36) = 147.5, take as 147
- 4. Sin(54) = 202.5, take as 202
- 5. Sin(72) = 237.5, take as 237
- 6. Sin(90) = 250
- 7. Sin(108) = 237.5, take as 237
- 8. Sin(126) = 202.5, take as 202
- 9. Sin(144) = 147.5, take as 147
- 10. Sin(162) = 77.5, take as 77

Ở đó, bây giờ mình đã tìm ra bảng sin: [0,77,147,202,237,250,237,202,147,77] Đó là 10 giá trị. Bạn có thể lặp lại quy trình cho bao nhiêu giá trị tùy ý. Bây giờ, làm thế nào để sử dụng điều này để thực hiện sóng sin: PR2 = 249 Ban đầu, CCPR1L = 0
mình cho phép ngắt CCP1, bằng cách ghi một ngắt vào CCP1IE và GIE và PEIE. Ngắt xảy ra

vào cuối mỗi chu kỳ, tức là khi TMR2 = PR2, tức là khi TMR2 đếm được 250 bước, tức là 6,25us, tức là ở tần số 16kHz – tần số sóng mang của mình.
mình đặt bảng sin dưới dạng một mảng. Hãy lấy biến là “sinval”.

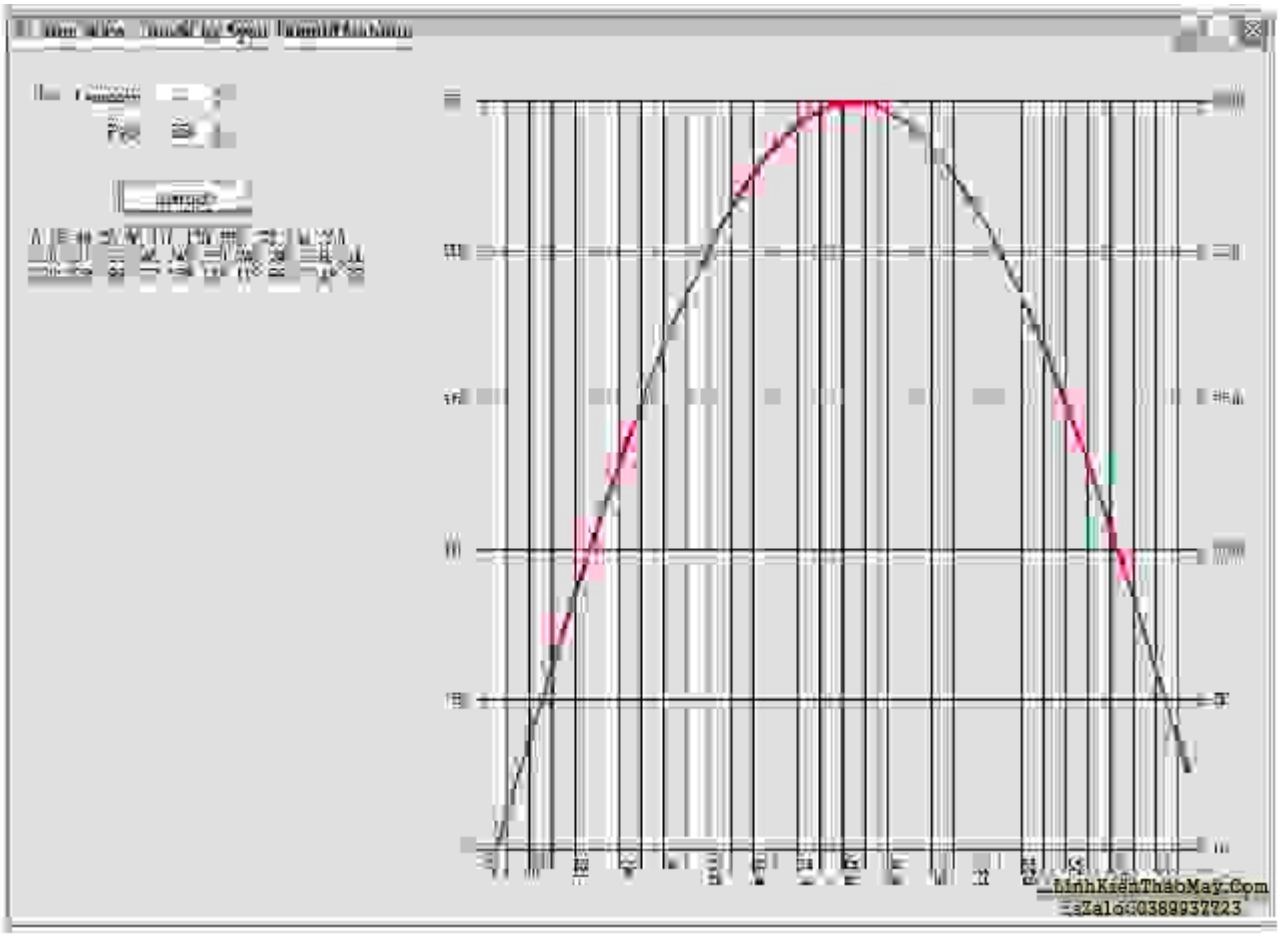
Vì vậy,
sinval = [0,77,147,202,237,250,237,202,147,77] mình sẽ lấy một biến để hoạt động như chỉ số mảng. Hãy coi đó là “ind”. Ban đầu, ind = 0. mình cũng sẽ sử dụng một biến khác báo hiệu sự kết thúc của chỉ mục. Hãy coi đó là “endi”. mình biết có 10 giá trị, sinval [0] thành sinval [9], vì vậy mình sẽ gán endi 10. Vì vậy, ban đầu, endi = 10
Trong ISR, trước tiên mình sử dụng ind để xác định vị trí hiện tại của mảng. Vì vậy, lúc đầu bao gồm. °. mình gọi sinval với chỉ số là ind. Vì vậy, khi bao gồm cả °, mình gọi sinval [0], là 0. mình chuyển giá trị này cho CCPR1L. Sau đó, mình tăng biến chỉ mục “ind” và so sánh biến này với “endi”. Ind không bằng endi, vì vậy mình tiếp tục. Lặp lại quy trình cho đến khi tất cả được thực hiện.
Vì vậy, khi ind = 9, sinval [9] được gọi. mình đã truy xuất giá trị 77. mình chuyển giá trị này tới CCPR1L. Sau đó, sau khi tăng dần, ind = 10. mình luôn so sánh giá trị với endi. Vì vậy, ind = endi, vì vậy, mình đảo ngược hướng của cầu đầy đủ bằng cách chuyển đổi bit 7 của ECCP1CON, là P1M1 hoặc EPWM1M1.
mình tiếp tục một lần nữa, lần này xóa ind, để mình có lại ind = 0 và khi ind = endi, mình đảo ngược hướng một lần nữa.



TRUNG TÂM SỬA CHỮA ĐIỆN TỬ QUẢNG BÌNH
MR. XÔ - 0901.679.359 - 80 Võ Thị Sáu, Phường Quảng Thuận, tx Ba Đồn, tỉnh Quảng Bình

Vì vậy, những gì mình đã đạt được là, lúc đầu QA và QD đang bật, mình điều chỉnh theo một hướng, sau đó tắt QA và QD và QB và QC theo hướng khác. Nếu mình sử dụng bộ lọc LC, mình sẽ đạt được sóng hình sin. Nhưng…. Có một điều còn lại. Tần số không phải là 50Hz. Tại sao?
Tần số sóng mang của mình là 16kHz và trong một sóng tổng, mình có 20 mẫu. Vì vậy, tần số là 16/20 = 0,8 kHz = 800Hz.

800Hz là tần số. Vì vậy, chu kỳ là 1,25ms. Đối với 50Hz, mình cần khoảng thời gian là 2Oms. Vì vậy, mình phải kéo dài thời gian của mình (20 / 1,25) = 16 lần. Điều này có thể được thực hiện, bằng cách gửi mỗi mẫu 16 lần, do đó kéo dài khoảng thời gian và do đó, mình có tần số 50Hz.
Điều này có nghĩa là gửi mỗi giá trị của sinval 16 lần trong ISR và sau đó tăng thanh ghi chỉ mục
Vì vậy sinval [0] sẽ được gửi 16 lần, sinval [1] sẽ được gửi 16 lần, sinval [2] 16 lần, v.v. Điều này có nghĩa là, đối với 16 ngắt, cùng một giá trị sẽ được chuyển đến CCPR1L và sau đó “ind” sẽ được tăng lên.
Để làm cho quá trình lấy bảng sin dễ dàng, mình đã viết một phần mềm trong VB có tên là “Sine Wave”. Điều này yêu cầu bạn phải cài đặt .NET framework. Đã cài đặt phiên bản .NET 3 mới nhất. Và bạn có thể dễ dàng có được một bảng sin. Mình sẽ upload phần mềm này miễn phí cho những ai muốn làm bàn sóng sin sử dụng.
Đây là ảnh chụp màn hình:


LinhKienThaoMay.Com
Zalo:0389937723

GENERATION AND IMPLEMENTATION OF SINE WAVE TABLE

SYED TAHMID MAHBUB

A sine wave is a wave with formula y=sin(x). Now we need to make this sine wave digitally, that is, "using square waves". Let's explain with an example.

When digitally producing sine wave, we do so using a fixed number of square wave pulses. The larger the number, the cleaner the sine wave. So, for example' sake, we'll use 10 samples, though, keep in mind, in practice, 10 is too less and usually at least 32 are taken.

So, we have y=sin(x). x varies between 0' to 360'. But, we'll only focus on one half and then reverse this to have the negative half. Since we have 10 samples, we divide the positive half of the sine wave into 10 parts. The first part is 0' to 18'. We'll send pulses, one at 0' and the other at 18', the next pulse at 36', then at 54' and so on. So, the values we need are:

1. Sin(0) = 0
2. Sin(18) = 0.31
3. Sin(36) = 0.59
4. Sin(54) = 0.81
5. Sin(72) = 0.95
6. Sin(90) = 1
7. Sin(108) = 0.95
8. Sin(126) = 0.81
9. Sin(144) = 0.59
10. Sin(162) = 0.31

We don't take sin(180) as this value will be next called when we implement sin(0), which both equal 0.

Now, we know how to find the values. Now, how to implement this. So, we're using PIC PWM. 1 means 100% duty cycle, 0 means 0% and 0.5 means 50%. So, from the above table, we know that:

1. Sin(0) = 0%
2. Sin(18) = 31%
3. Sin(36) = 59%
4. Sin(54) = 81%
5. Sin(72) = 95%
6. Sin(90) = 100%
7. Sin(108) = 95%
8. Sin(126) = 81%

LinhKienThaoMayCom
Zalo0389937723

9. Sin(144) = 59%
10. Sin(162) = 31%

Now, we need to put these values into a proper sine table. For that we need to know the value of PR2.

PWM Period = [(PR2) + 1] * 4 * TOSC * (TMR2 Prescale Value)

We know, PWM period, TOSC and prescale value, so putting this and rearranging gives us,

PR2 + 1 = PWM Period / (4* TOSC * {TMR2 Prescale Value})

Therefore,

PR2 = [PWM Period / (4* TOSC * {TMR2 Prescale Value})] – 1

*Formula derived from that given in datasheet.

Now, we need to decide at which frequency we are going to do the PWM (carrier frequency). Say, we have an oscillator of 16MHz and we decide to use a carrier frequency of 16kHz. Then, if you put in the values into the above formula, we have:

PR2 = [(1/16,000)/(4 * {1/16,000,000} * 1] – 1

Giving us,

PR2 = 249

So we assign 249 to PR2. The duty cycle register is CCPR1L. When CCPR1L = 0, duty cycle = 0, when CCPR1L = (PR2 + 1), duty cycle = 100%. This is because (PR2 + 1) = Period

So, CCPR1L/(PR2 + 1) = Duty cycle

So, the table:

1. Sin(0) = 0%
2. Sin(18) = 31%
3. Sin(36) = 59%
4. Sin(54) = 81%
5. Sin(72) = 95%
6. Sin(90) = 100%
7. Sin(108) = 95%
8. Sin(126) = 81%
9. Sin(144) = 59%
10. Sin(162) = 31%

LinhKiệnThảoMáyCom
ZaZalo:0389937723

Can be rearranged as:

1. Sin(0) = 0 * 250
2. Sin(18) = 0.31 * 250
3. Sin(36) = 0.59 * 250
4. Sin(54) = 0.81 * 250
5. Sin(72) = 0.95 * 250
6. Sin(90) = 1 * 250
7. Sin(108) = 0.95 * 250
8. Sin(126) = 0.81 * 250
9. Sin(144) = 0.59 * 250
10. Sin(162) = 0.31 * 250

Here we multiplied by 250 as 250 is the peak of the sine wave [y=sin(x) * 250] as 250 is the period.

So we have:

1. Sin(0) = 0
2. Sin(18) = 77.5, take as 77
3. Sin(36) = 147.5, take as 147
4. Sin(54) = 202.5, take as 202
5. Sin(72) = 237.5, take as 237
6. Sin(90) = 250
7. Sin(108) = 237.5, take as 237
8. Sin(126) = 202.5, take as 202
9. Sin(144) = 147.5, take as 147
10. Sin(162) = 77.5, take as 77

There, we have now found out our sine table:

[0,77,147,202,237,250,237,202,147,77]

That's 10 values. You can repeat the process for as many values as you need.

Now, how to use this for implementation of sine wave:

PR2 = 249

Initially, CCPR1L = 0

We enable CCP1 interrupt, by writing one to CCP1IE and GIE and PEIE. The interrupt occurs at the end of every period, ie, when TMR2 = PR2, ie, when TMR2 counts 250 steps, ie 6.25us, ie at a frequency of 16kHz – our carrier frequency.

We set the sine table as an array. Let's take the variable as "sinval".

LinhKienThaoMay.Com
Zalo:0389937723

So,

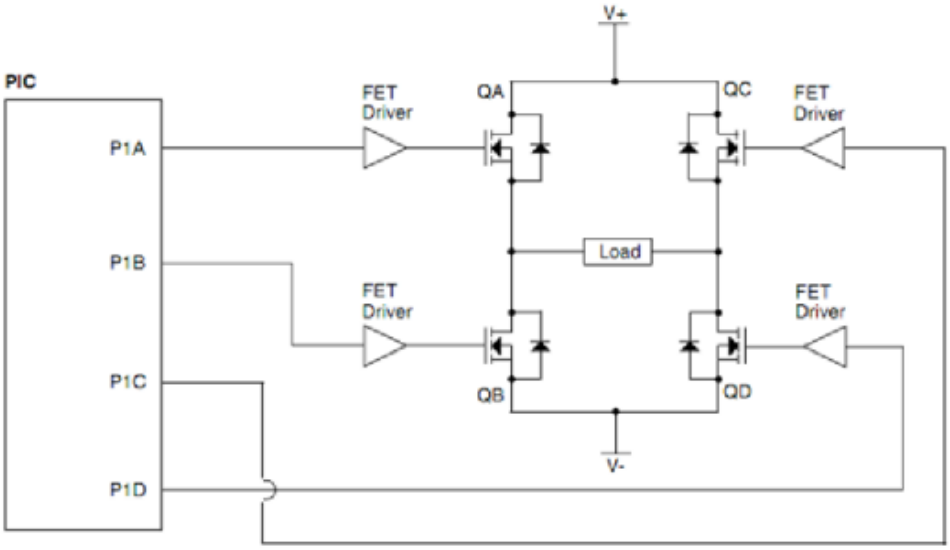
sinval = [0,77,147,202,237,250,237,202,147,77]

We'll take a variable to act as the array index. Let's take that as "ind". Initially, ind = 0. We'll also take another variable that signals the end of the index. Let's take that as "endi". We know there are 10 values, sinval[0] to sinval[9], so we'll assign endi 10. So, initially, endi = 10

In the ISR, we first use ind to determine the current position of the array. So, at first ind=0. We call sinval with the index being ind. So, when ind=0, we call sinval[0], which is 0. We pass this value to CCPR1L. Then, we increment the index variable "ind" and compare this against "endi". Ind does not equal endi, so we continue. Repeat the procedure until all are done.

So when ind = 9, sinval[9] is called. We have retrieved the value 77. We pass this to CCPR1L. Then after incrementing, ind=10. We always compare the value against endi. So ind = endi, so, we reverse the direction of the full bridge by toggling bit 7 of ECCP1CON, which is P1M1 or EPWM1M1.

We continue again, this time clearing ind, so that we have ind = 0 again, and when ind = endi, we reverse the direction again.



So, what we have achieved is, while at first QA and QD are on, we modulated in one direction, and then with QA and QD off and QB and QC on the other direction. If we use LC filter, we'll achieve a sine wave. But.... There is one thing remaining. The frequency isn't 50Hz. Why?

Our carrier frequency is 16kHz and in a total wave, we have 20 samples. So, the frequency is 16/20 = 0.8 kHz = 800Hz.

LinhKienThaoMay.Com
Zalo:0389937723

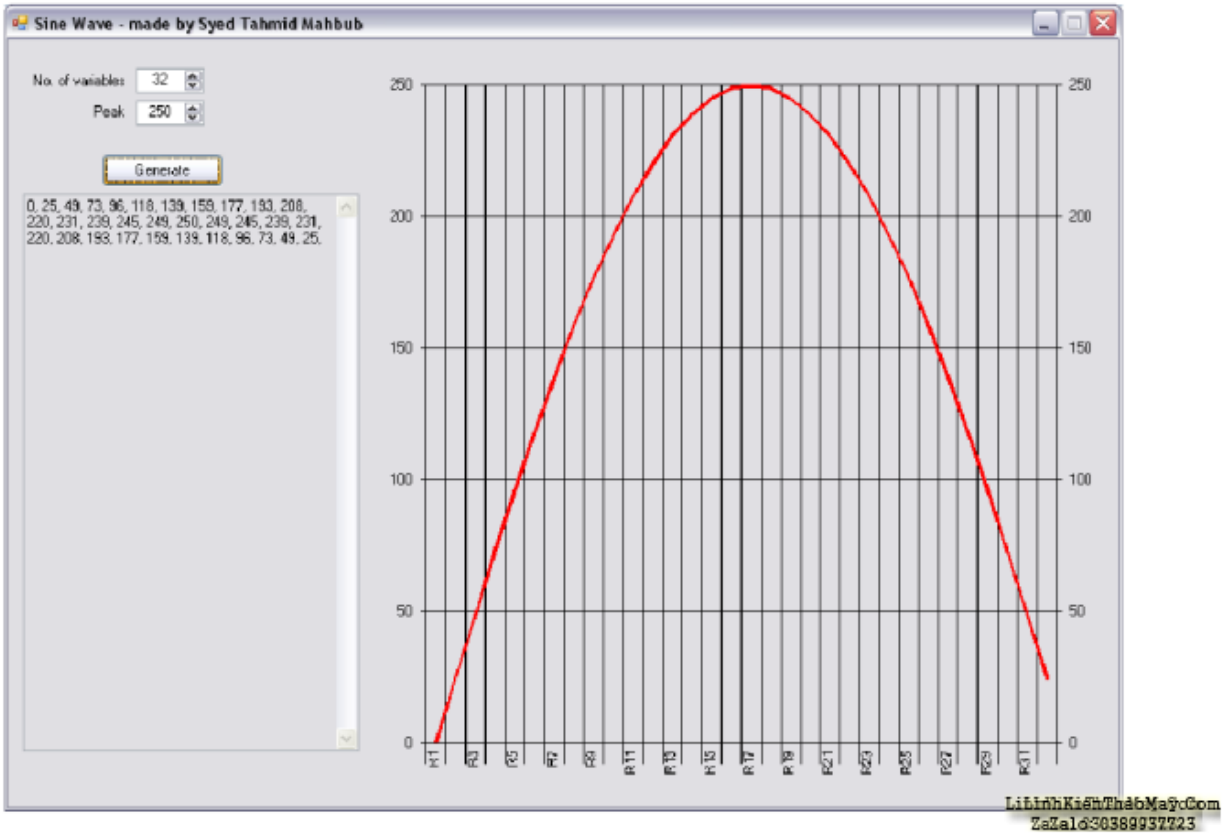
800Hz is the frequency. So period is 1.25ms. For 50Hz, we need a period of 20ms. So, we must prolong our period (20/1.25) = 16 times. This can be done, by sending each sample 16 times, thus prolonging the period and so, we have a 50Hz frequency.

This would mean sending each value of sinval 16 times in the ISR and then increment the index register "ind".

So sinval[0] would be sent 16 times, sinval[1] would be sent 16 times, sinval[2] 16 times and so on. This means, for 16 interrupts, the same value would be passed to CCPR1L and then "ind" would be incremented.

To make the process of obtaining the sine table easy, I wrote a software in VB called "Sine Wave". This requires you to have .NET framework installed. Have the latest version of .NET 3 installed. And you can easily get a sine table. I will upload this software for free for those who want to make sine wave table for their use.

Here's a screenshot:



Các bài viết tương tự:

1. CÁCH TẠO 1 TRANG WEB – mình muốn tạo 1 trang web ai biết xin chỉ cách lập
2. Điều khiển nguồn AC bằng PWM sử dụng MOSFET/IGBT
3. Điều khiển tốc độ động cơ DC bằng PWM sử dụng vi điều khiển 8051
4. Feedback trong Inverter sine (dựa trên dòng PIC16F*)
5. Inverter sine 1000W bằng EGS002 và biến áp sắt 12V (2 dây)
6. Inverter sine 800W – biến áp xung EE42 bằng EGS002
7. lò vi sóng – panasonicinverter – ko chay dc ở chế độ vi sóng, khi e bật ở chế độ vi sóng thì chạy dc 30 giây máy ngắt kêu tít tít và dừng lại. đồ đặt ở trong lò không thấy nóng. e đã thay cả 3 vỉ: vỉ nguồn, vỉ khiển,vỉ inverte, thay nốt cả súng mà máy vẫn ko chạy, vẫn bị như cũ

8. Mạch inverter sine đơn giản dễ chế tạo nhất
9. Năm con Gà, chúc mọi người vui vẻ như Chim Sẻ, khỏe mạnh như Đại Bàng, giàu sang như chim Phụng, làm lụng như chim Sâu, sống lâu như Đà Điểu. – Năm con Gà, chúc mọi người vui vẻ như Chim Sẻ, khỏe mạnh như Đại Bàng, giàu sang như chim Phụng, làm lụng như chim Sâu, sống lâu như Đà Điểu.
10. Sửa nguồn ATX chạy 3843 mất áp IC PWM do hư trở mồi
11. Tạo sóng sin sử dụng xung SPWM trong PIC16F684
12. Thiết kế INVERTER SINE 600w bằng TDS2285 và EE55 hoặc PC40