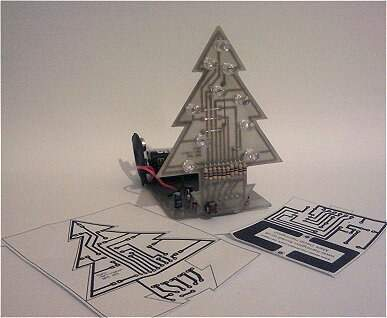
Simple Christmas Tree project aimed at all ages Ideal gift, Christmas decoration, or excellent beginners starter project Uses a random number generator to illuminate led’s   
Features:- Very few components, Easy to assemble and make Incorporates pic-micro technology

Shows you what you can do with a simple 12F675 and a bit of practice



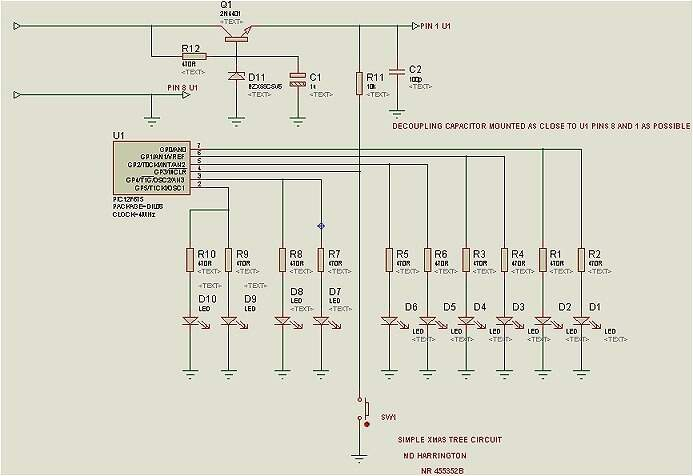
The circuit comprises a microchip 12F675 cpu which utilises GPIO port pins Gp0 to Gp2 and GPIO port pins Gp4 to Gp5 as output for our leds   
  
Gpio port pin 3 is for switched input for later use I have not implemented this as yet,but have incorperated this for pattern changing on the Xmas tree which I will endeavour to get round to within the current week

Q1,R12,D11,C1 form the 5volt supply for the microchip and leds

R1 to R10 provide current limiting for the leds and the microchip

C2 is decoupling for the micro as specified in the data-sheet

See Image On Next Page

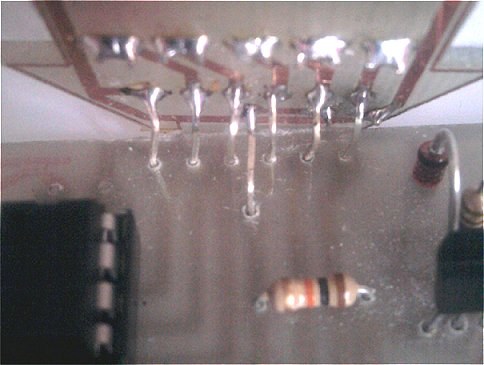


Christmas Tree Circuit boards



Image above shows the two printed circuit boards The right hand PCB forms the base of the completed project to which fix the tree PCB

Below how the tree fits together with the base pcb which holds our micro chip plus PP3 9 volt battery so that the unit stands up



All circuit boards were made by hand using Press and Peel available from ebay or [www.ronlin.co.uk](http://www.ronlin.co.uk/)

I’ve always found this to provide excellent results even down to surface mount flat pack level, although it does require patience and plenty of practice



Code written using proton Basic   
  
Please see code below

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\* Name : XmasTree.BAS \*

'\* Author : MD Harrington \*

'\* Notice : Copyright (c) 2012 @www.harrington.force9.co.uk \*

'\* : All Rights Reserved\*

'\* Date : 02/12/2012 \*

'\* Version : 1.0 \*

'\* Notes : Check that CP\_ON is set to off if experimenting

'\* : and recompiling before blowing chip \*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Device 12F675

Config CPD\_OFF , WDT\_OFF , BODEN\_OFF ,PWRTE\_OFF,MCLRE\_OFF ,INTRC\_OSC\_NOCLKOUT,CP\_ON

Xtal = 4

'START PROGRAM

On\_Interrupt ISRoutine

Symbol T0IF = INTCON.2 ' TMR0 Overflow Interrupt Flag

Symbol T0IE = INTCON.5 ' TMR0 Overflow Interrupt Enable

Symbol GIE = INTCON.7 ' Global Interrupt Enable

Symbol PS0 = OPTION\_REG.0 ' Prescaler Rate Select

Symbol PS1 = OPTION\_REG.1 ' Prescaler Rate Select

Symbol PS2 = OPTION\_REG.2 ' Prescaler Rate Select

Symbol PSA = OPTION\_REG.3 'Prescaler Assignment

Symbol TOCS = OPTION\_REG.5

Symbol T0SE = OPTION\_REG.4

Dim mnumber As Word ' for storing our random number

Dim mindex As Byte ' variable used in for next loops

Dim mcount As Byte ' every time our timer overflows we use this to add 1

Dim SWVAL As Byte ' SWVAL stores of stwich is high or low

Dim FLAGS As Byte ' flags register is for various bool flags that we set , reset

Dim mpressedcnt As Byte ' stores the number of times button is pressed

Dim PATTERN As Byte ' pattern is our pattern detect register what pattern to use

Dim LHSindex As Byte ' for scanning the tree up or down

Dim RHSindex As Byte

Dim RHValue As Byte ' the value we are going to shift

Dim LHValue As Byte

Symbol SW1 = GPIO.3 ' SW1 is atteched to GPIO.3

Symbol mpressed = FLAGS.0 ' Flag to indicate if button was pressed

Symbol msecflag = FLAGS.1 ' Flag to indicate 2/15th second elapse completed

GoTo Main

'\*\*\*\*\*\*\*\*\* InterruptServiceRoutine start here \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ISRoutine:

Context Save

' disable the interrupt

GIE = 0 ' disable global interrupts

T0IE = 0 ' disable the timer flag

T0IF = 0 ' clear the interrupt flag

' since the timer overflows every 15 times per second

' we add 1 for each 2/15th

' so we scan the switch every 2/15 overflows

If mcount < 1 Then

mcount = mcount +1

msecflag = 0

Else

mcount = 0 ' reset if we have

'2 in every 15 overflow complete

msecflag = 1 ' set to true

'we have one second lapse

End If

'scan the switch if msecflag is true

If msecflag = 1 Then ' one second has elapsed

'Scan switch

If SW1 = 0 Then

While SW1= 0

' wait for button release

Wend

mpressed = 1 ' button press was detected

' no button press was detected

End If

' if we detected button press

'then we shift the pattern left for each

If mpressed = 1 Then

'is the button pressed counter less than two

If mpressedcnt < 2 Then

PATTERN = PATTERN << 1 ' Yes counter was < 2

' shift our pattern

'select register 1 left

mpressedcnt = mpressedcnt + 1

Else

PATTERN = 1 'reset pattern register

mpressedcnt = 0 ' reset button pressed counter

End If

End If

mpressed = 0

' re enable the interrupt

TMR0 = 0

T0IE = 1 ' enable the timer flag

GIE = 1 ' enable global interrupts

Context Restore

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* end interrupt procedure \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main program starts here \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Main:

GoSub Init

TMR0 = 0

' start the interrupt off

GIE = 1 ' enable global interrupts

T0IE = 1 ' enable the timer flag

Loop:

Select PATTERN

Case 1

GoSub COUNT0

Case 2

GoSub COUNT1

Case 4

GoSub COUNT2

Case Else

End Select

GoTo Loop:

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Main program ends here \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

COUNT0:

GPIO = %00111111 ' switch outputs on switch off

DelayMS 1000

GPIO = 0

DelayMS 2000

For mindex = 0 To 2 Step 1

GPIO = %00111111

DelayMS 50

GPIO = 0

DelayMS 50

Next mindex

Return

COUNT1:

MYRAND:

mnumber = Random ' get a random number

While mnumber > 63 Or mnumber < 7

mnumber = Random

Wend

' because we are only dealing with 5 bits

' we look for a random number

' between 0 and 64 i.e 1 to 63

' if random number is outside this

' produce another random number

If mnumber > 0 And mnumber < 64 Then

For mindex = 0 To 5 Step 1

' assign gpio to the randon number produced

GPIO = mnumber

DelayMS 30

GPIO = 0

DelayMS 30

Next mindex

mnumber = ~ mnumber

DelayMS 1000

For mindex = 0 To 5 Step 1

' assign gpio to the randon number

GPIO = mnumber

DelayMS 30

GPIO = 0

DelayMS 30

Next mindex

End If

Return

COUNT2:

If LHSindex = 0 Then

mnumber = %00111111

GPIO = mnumber

End If

If LHSindex < 7 Then

GPIO = mnumber

mnumber = mnumber ^ LHValue ' Xor the values

LHValue = LHValue << 1 ' shift left by one

LHSindex = LHSindex + 1 ' increment index

Else

LHSindex = $00

LHValue = $01

End If

DelayMS 50

Return

Init:

' sets up the OSCCAL calibration for this device

Asm

Bsf STATUS,RP0

Call 0x3ff

Movwf OSCCAL

Bcf STATUS,RP0

EndAsm

VRCON = $00

GPIO = $00 ' clear the ports

WPU = $00 ' weak pull up registers

CMCON =$07

INTCON = $00

ANSEL = $00

TRISIO = $00

' variable Initialisation

mcount= $00

mpressed = 0

PATTERN = $01 ' set pattern register to 1

'for our first pattern to be displayed on the xmas tree

mpressedcnt = $00 ' set to zero to start with

RHValue = $20

LHValue = $01

' select 1:256 timer ratio Timer will overflow 15 times per second

PS0 = 1

PS1 = 1

PS2 = 1

PSA = 0 ' prescaler assigned to timer 0 module

TOCS = 0 ' Timer mode is selected

T0SE = 0 ' select the rising edge bit

DelayUS 50

Return

Merry Xmas to all members and followers of this site Lets hope next years brings peace and prosperity to all

Mark