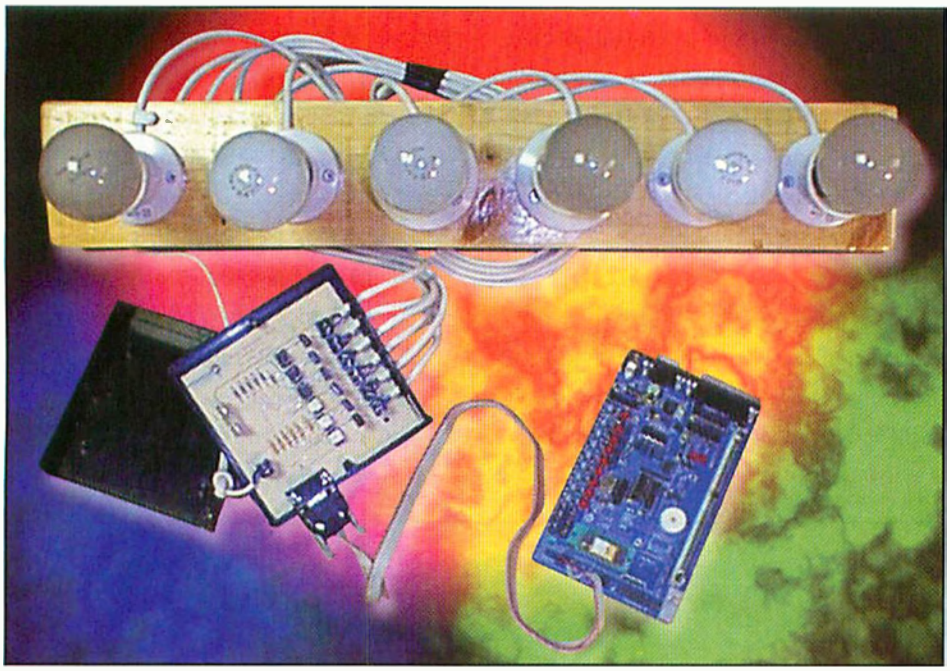
DISCO Light Controller

An ST7 based microcontroller 6 channel lighting sequencer by Tony Ashton



Introduction

This article is concerned with looking at a method of controlling 6 x 100 watt disco lights operating at 240Volts AC. The project will use minimal hardware and allow the user to update the sequence of the lights within the club environment. These disco lights are controlled by the Kanda ST7 Starter Kit, or flash circuit, which is shown in Appendix 2 later on in this article, together with the use of another circuit that will allow the operation of the mains supply voltage. The Kanda ST7 Starter Kit was designed to provide maximum flexibility to users of the ST7 family of high-performance microcontrollers. The Kanda Starter Kit offers many features such as using the 16-bit timers to produce Pulse Width Modulation (PWM) control. The project was designed around the ST72254G2 flash Integrated Circuit (IC) which is a 32-pin flash device and will allow the code to be updated easily when in operation. The Kanda ST7 Starter Kit, with a flash device, or an alternative circuit based around the ST72254G2 device will be connected to a power controller circuit in the way shown in this article, in order to provide control of the disco lights.

Figure 1 shows how the Kanda ST7 Starter Kit or an equivalent circuit that incorporates the ST72254G2 flash device is connected to the controller board. A lead made up of seven wires will be taken from the Kanda ST7 Starter Kit to the power controller board. Six of these wires will be connected from PB2-PB7 and the other to VDD, of the Kanda ST7 Starter Kit in order to provide 5 volts for the triac driver circuits . Each of these seven wires will be taken to a 9-way 'D' type connector on the power controller board, this is shown in Figure 2.

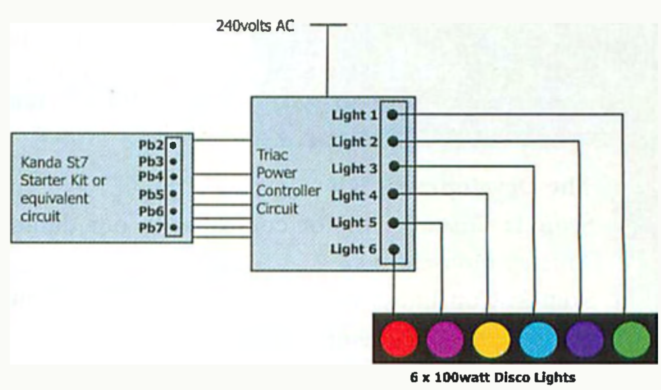


Figure 1. Block diagram of system

Power Controller Circuit

The power controller circuit uses triacs, and optocoupler triac drivers, the circuit diagram of the power controller board can be seen in Figure 2.

Figure 2 uses the Z0410DF triac, and the MOC3020 optocoupler triac driver which when combined together will provide smooth power control. A common problem within power switching circuits is that triacs switch from off state to the on state in a very fast time, and can produce extremely rapid increases in load current. Such a current step will produce harmonics within the circuit and can cause interference to AM radio reception. A simple way to prevent this would be to incorporate a capacitor and an inductor in the schematic to form a low pass filter and hence prevent the harmonic energy from reaching the load wiring and radiating everywhere. This filter will therefore reduce the interference to a nearby AM radio receiver. However the above circuit uses the MOC3020 optocoupler triac driver and can be used as a good noise barrier for digital signal isolation. The ground loop is also broken and any noise picked up by the cabling appears as a common mode noise at the terminals, and hence can be easily cancelled either solely by the optocoupler or using additional common mode chokes, thus eliminating the need for the low pass filter.

Protection Tips

The project has been designed to be used with 6 x 100 watt light bulbs, at a working voltage of 240 volts however, great care must be taken when working with this type of voltage. When the circuit has been produced and built it is recommended that it is inserted into a hard plastic case, and that all wires entering and leaving the box should be securely fixed, and that no bare wires can be seen. After the circuit has been built and checked for visual defects, the circuit was tested using a continuity tester in order to detect any

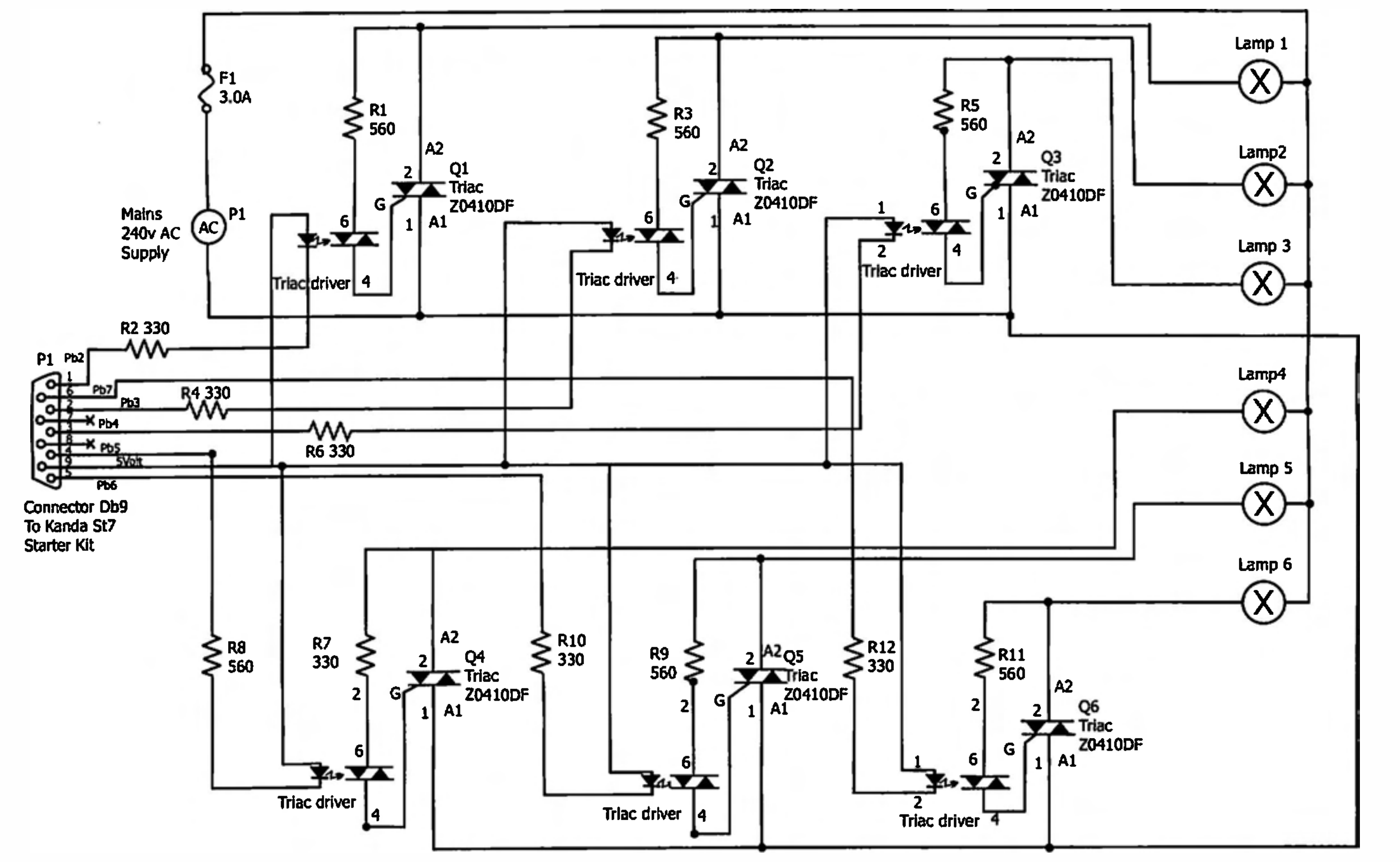


Figure 2. Schematic diagram of the power Controller Circuit

short circuits that might have arisen. When the circuit is running, the box must be kept in a closed position and nothing must be inserted in to the box. If any problems arise make sure that the power is disconnected from the mains supply before investigations start. A 3.0 Ampere fuse must be fitted to the circuit when working with 6 x 100 light bulbs operating a 240 volts AC.

User Instructions

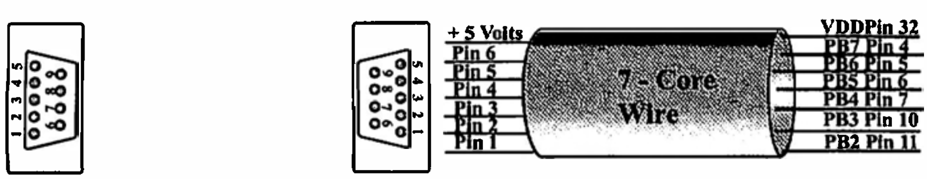
Before the power controller circuit is built make sure that the dimensions of the PCB will fit securely into the plastic case, and that fixing screws can be inserted into the board without short circuiting the PCB. The 9-way 'Male' 'D' type socket can be fitted to the PCB ensuring that it is placed close to the side of the box. The side of the plastic case can then be cut out in order for the 'Male' 'D' type socket to be accessed. It should also be ensured that the six 2-way terminal blocks for the connection of the lights are towards the front of the plastic case. The plastic case can then be drilled out in order for the cable of the disco lights to sit easily in the terminal blocks. The same procedure should be carried out for the mains cable which can be placed at the back of the circuit. If the cables are loose in the drilled holes then a grommet and cable clamps should be fitted for added protection. A seven way cable is then required with the 9-way 'Female' 'D' type socket connected to one end, and seven single in-line sockets connected to the other, these seven will then go to PB2-PB7 (pins 4-7, and 10-11), and VDD (pin 32) of the Kanda ST7 Starter Kit respectively. A diagram showing the connections can be seen in Figure 3.

Once the board has been tested and all the connections are secure, power can be supplied to the board. The initial start off will set the sequences to 15%, which will be the fastest sequence of the lights. However the Kanda ST7 Starter Kit can control the speed settings of the sequence of lights by simply

9-way 'Male' 'D' type socket connected to power controller board.

9-way 'Female' 'D' type socket.

Connections to Kanda ST7 Starter Kit Board.



9-Way 'Male'	9-Way 'Female'	Connection
Pin 1	Pin 1	PB2
Pin 2	Pin 2	PB3
Pin 3	Pin 3	PB4
Pin 4	Pin 4	PB5
Pin 5	Pin 5	PB6
Pin 6	Pin 6	PB7
Pin 7	Pin 7	NC
Pin 8	Pin 8	NC
Pin 9	Pin 9	+ 5 Volts

Connections from Kanda ST7 Starter Kit to Power Controller Board.

Figure 3.

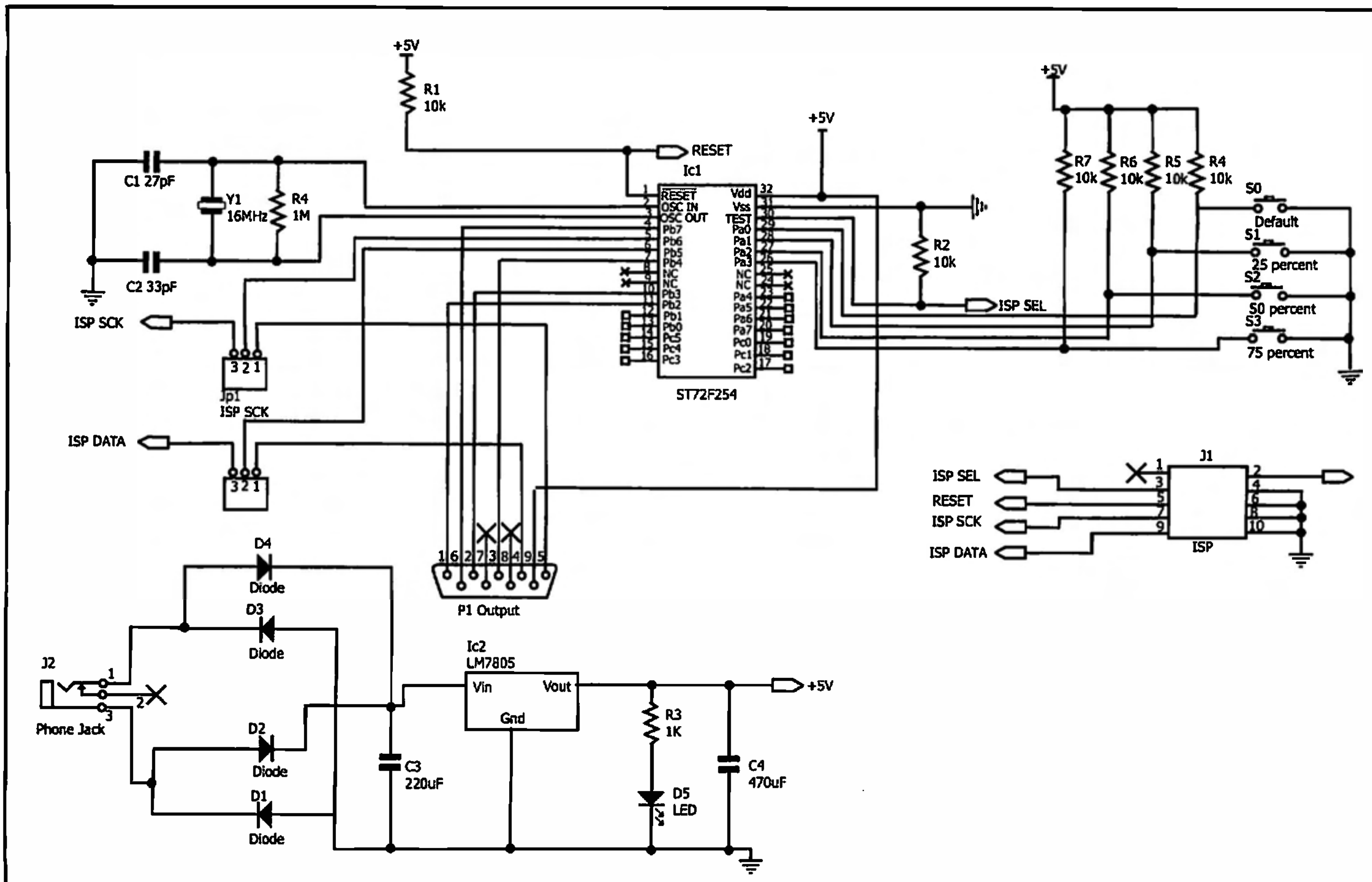


Figure 4. Schematic Diagram for ST72254 Flash Circuit with ISP.

pressing keys 0 - 3, the settings are shown below:

Button '0'	Sets Sequence to '15%'
Button '1'	Sets Sequence to '25%'
Button '2'	Sets Sequence to '50%'
Button '3'	Sets Sequence to '75%'

If button 3 on the Kanda ST7 Starter Kit is pressed the sequence of lights will operate at 75% which is the longest time interval between on/off of the lights. So this means that for the whole period the lamp is on for three-quarters of the time and off for the other quarter. At any time the user can select another speed such as button 2 and the sequence of the lights will increase to 50% and decrease the on/off time of the lights.

Software Design

The software was designed using the ST7 Starter Kit Assembler code. The software listings generated can be seen in Appendix 1 of this article. It will be noticed that there are three stages to the software:

The .sts which is the Set-up code
The .stv which is the Vector code
The .stm which is the Main code

The Set-up file will generate all the code necessary to set the ports of the ST7 microcontroller as inputs or outputs. The set-up code is also used to set the timers, and enable the interrupts. Once the interrupts have been enabled they can be manipulated within the vector file. The reset vector information can also be found in the vector file. The main file will then be left so as the user can input the main body of the code. Using the three different files in the way explained above keeps the code clear and easy to follow. If changes have to be made to the code at a later date it will be easy for the user to find the correct place to insert the new code.

In order to generate the necessary time delays between the flashing lights, the programmable timer was set-up to be a 16-bit free running increasing counter with its associated 16-bit registers. The two read-only 16-bit registers contain the same values but with the difference that reading the ACLR (Alternate Counter Low Register) does not clear the timer overflow flag. However writing in the CLR (Counter Low Register) or ACLR resets the free running counter to the FFFCh value. The timer clock depends on the clock control bits of the CR2 register. The value in the counter register repeats every 131.072, 262.144 or 524.288 internal processor clock cycles depending on the CC1 and CC0 bits. When the program is entered the counter is set to the default value of 15 percent; this is achieved by the following code.

```
Set:    ld a,#$26
        ld TAOC1HR,a
        ld a,#$66
        ld TAOC1LR,a
        clr TAACLR
```

The code shown above is used to set the default value for the counter in the Timer A Output Compare 1 Register (TAOC1R). When the code is running the value 2666h appears in the TAOC1R. The last line of the above code is used to set Timer A Alternate Counter Register (TAACR) to FFFCh. This is an increasing counter which will proceed to 0000h on the next clock cycle after FFFFh. When the value in TAACR matches the value in TAOC1R (in this case both set to 2666h) an interrupt is generated. This starts the sequence of lights at the default speed and will continue at this setting until buttons 0-3 on the Kanda ST7 Starter Kit are pressed which will change the speed of the sequence of lights. The code was designed to allow the user to update the sequence of lights within the vector file, which can then easily be transferred to the flash device by using a keyfob.

continued on page 32

Set-up Code

```
;ST7 Setup File - Created by Application Builder
; Setup File Created with Device Set to ST72C254G2
.Include $DeviceName        ; Include Device Definition File
(ST72C254G2.str)

        segment 'Rom'    ; Code segment - Address defined in
Definition file (.str)

Start:          ; Reset Label

                ;**********Timer A Setup Code
            ld a,#$40
            ld TACR1,a
            ;**********Timer A setup code
            ld a,#$80
            ld TACR2,a
            ;*Timer A Output Compare Register 1
            ld a,#$80
            ld TAOC1HR,a
            ld a,#$00
            ld TAOC1LR,a

            ; ********* Port A Setup Code
            ld a, #$00          ; Bit On = Output, Bit Off
= Input
            ld PADDR, a         ; Port A Direction
Register
            ld a, #$FF          ; Option Selects Input or
Output Type
            ld PAOR, a          ; Port A Option Register

            ; ********* Port B Setup Code
            ld a, #$FF          ; Bit On = Output, Bit Off
= Input
            ld PBDDR, a         ; Port B Direction
Register
            ld a, #$00          ; OCMP1A,
            ld PBOR, a          ; Port B Option Register

            ; ********* Control Registers
            ld a, #$D9          ; Main Clock Out Off
            ld MISCR1, a        ; Slow Mode : Fcpu =
4000000Hz
             ; EI0 Interrupt on Falling and Rising Edge
Disabled
            ld a, #$00          ; Clock Filter Interrupt

            ld CRSR, a

            ld a, #$7F          ; Stack Pointer Low Byte
            ld s, a   ; Stack Pointer High Byte set by
hardware
            ld a, #$7F          ; Watchdog Control
Register
            ld WDGCR, a         ; Watchdog Disabled

            rim       ; Global Interrupt Enable
```

Vector File Code

```
; ST7 Vector File - Created by Application Builder

        ; ********** Interrupt Service Routines
        ; File Created with Device Set to ST72C254G2
        segment 'Rom'    ; Code segment - Address defined in
Definition file (.str)
TimerA:  ; ******* TimerA Interrupt ISR Code Here
;**********************************************
;       CHANGING THE SEQUENCE BELOW WILL     *
;       ALTER THE SEQUENCE OF THE LIGHTS     *
;       IN ORDER TO UPDATE AT THE CLUB             *
;**********************************************
TAOF:           ; Test TOF Flag (TASR.5)
        nop
        call seq_1
        jra TAOF
        nop
        iret

seq_1:  call light_1
        call light_2
        call light_3
        call light_4
        call light_5
        call light_6
        call light_5
        call light_4
        call light_3
        call light_2
        ret
;*****************************************
;       ADD LIGHT SEQUENCE HERE !!!!         *
;*****************************************
light_1: nop
                    bres pbdr,#2
                          call tim
                    bset pbdr,#2
                    ret

light_2: nop
                    bres pbdr,#3
                          call tim
                    bset pbdr,#3
                            ret

light_3: nop
                    bres pbdr,#4
                    call tim
                    bset pbdr,#4
                    ret

light_4: nop
                    bres pbdr,#5
                    call tim
                    bset pbdr,#5
                          ret

light_5: nop
                    bres pbdr,#6
                    call tim
                    bset pbdr,#6
                    ret

light_6: nop
                    bres pbdr,#7
                    call tim
                    bset pbdr,#7
                          ret

tim:                nop
                    call timer
                    ld a,#$3F
                    ld stay,a
                    nop
                    ret

          ; ******* End Timer A Interrupt
                    segment 'vectit'     ; Vector segment -
Address defined in Definition file (.str)

                    DC.W    0
                    DC.W    0
                    DC.W    0
                    DC.W    0
                    DC.W    0
                    DC.W    0
                    DC.W    0
                    DC.W    0
                    DC.W    0
TimA:               DC.W    TimerA
                    DC.W    0
                    DC.W    0
                    DC.W    0
E0:                 DC.W    0
                    DC.W    0
reset:              DC.W    Start

                    end
```

Main Source Code

```
;ST7 Main File - Created by Application Builder
 ; Author : Tony Ashton
 ; Company : Kanda Systems Ltd
 ; Comment : Code for Disco Lights/Easy to update in club
environment

; File Created with Device Set to ST72C254G2
.include $SetupFileName    ; Set up file (.sts) is included
here
segment 'Ram0'  ; RAM Page Zero segment - Always at ADDRESS
$80-$FF
;Define user variables here if required - Example Syntax is
;COUNTER DS.B 1
;COUNTER2 DS.W 1
;PIN_NUMBER EQU 6

comp dc.b $00
stay dc.b $3F
        segment 'Rom'  ; Code segment - Address defined in
device file (.str)
set:                nop               ;default setup code
                    ld a,#$26
                    ld TAOC1HR,a
                    ld a,#$66
                    ld TAOC1LR,a
                    clr TAACLR        ;timer A alternate counter
low register
                    jra main

main:               ld a,#$FF;stays here until interrupt on port
a is generated
                    ld pbdr,a
                    nop
                    nop
                    jra main ;returns to main until interrupt is
detected

timer:              nop
                    btjf padr,#0,default
                    btjf padr,#1,timer_25
                    btjf padr,#2,timer_50      ;sets speed of
lights from 25-75 percent
                    btjf padr,#3,timer_75
                    jra d1

d1:                 clr TAACLR                 ;sets TACR to
FFFD
                    nop
                    nop
                    ld a,TAOC1LR      ;loads value into
accumulator
d2:                 cp a,TAACLR
                    jrule high
                    nop
                    jra d2
high:               ld a,TAOC1HR
high_1:  cp a,TAACHR
                    jrule go ;jumps to go if TAACHR lessthan or
equals to zero
                    nop
                    jra high_1        ;returns to compare if
TAACHR is greaterthan zero
go:                 nop
                    ld a,stay
                    cp a,#$00
                    jrule go_1        ;ends if stay is lessthan
or equal to zero
                    dec stay ;decrements value in order to reach
end of sequence
                    jra d1            ;jumps to d1 if stay is
greaterthan zero
go_1:               ret               ;jumps out of sequence and
turns on another light
;**********************************
;*                  TIMING INTERVALS  *
;**********************************
default: ld a,#$26         ;timing interval for 15 percent
t3:                 btjf padr,#0,t3            ;debounce
                    ld TAOC1HR,a
                    ld a,#$66
                    ld TAOC1LR,a
                    jra d1                     ;returns to d1

timer_25:ld a,#$3F         ;timing interval for 25 percent
t0:                 btjf padr,#1,t0            ;debounce
                    ld TAOC1HR,a
                    ld a,#$FF
                    ld TAOC1LR,a
                    jra d1                     ;returns to d1

timer_50:ld a,#$7F         ;timing interval for 50 percent
t1:                 btjf padr,#2,t1
                    ld TAOC1HR,a
                    ld a,#$FF
                    ld TAOC1LR,a
                    jra d1

timer_75:ld a,#$BF         ;timing interval for 75 percent
t2:                 btjf padr,#3,t2
                    ld TAOC1HR,a
                    ld a,#$FF
                    ld TAOC1LR,a
                    jra d1

        ; Interrupts in use, so .STV File called here
        ; This file contains shell Interrupt Service Routines
(ISR)
        ; and Interrupt Vector Table.  Add Your ISR Code to
this file

          .include $VectorFileName  ; Interrupt and Vector
file (.stv) included here
                    end
```

PARTS LIST

Kanda ST7 Starter Kit or Equivalent
Circuit Using ST72254G2 Device
ST72254G2 Integrated Circuit
6 x 6 pin dual-in-line sockets
6 x MOC3020 Optocoupler, Triac Drivers
6 x Z0410DF 4 Amp 400 Volt Triacs
1 x 9-way 'Male' 'D' Connector
1 x 9-way 'Female' 'D' Connector
6 x 330R Resistors
6 x 560R Resistors
7 x 2-way terminal blocks
1 x Fuse Holder
1 x 2.6 Amp Fuse
6 x 100 Watt 240 volts Disco Lights
1 x Plastic Hard Case to Cover PCB.

Conclusion

This project is based around the ST72254G2 flash device, which can be used with the Kanda ST7 Starter Kit, or alternatively a circuit can be built dedicated to this project based around the ST72254G2 device. A circuit showing a simple design for the flash device is given in Appendix 2. The project uses the flash device to control the operation of disco lights, which operate at 240 volts ac. The only other circuit that was used within the project was the power controller board, which was used to supply the necessary voltage to the disco lights. This circuit was based around a triac, and triac driver that provide complete isolation from the Kanda ST7 Starter Kit. The circuit will operate up to six disco lights moving in what ever sequence the user requires. Using the flash device gives an added advantage of updating the software within the club environment using one of Kanda's keyfobs. This will allow the code to be changed to develop a new sequence for the disco lights, and update them simply by plugging the keyfob into the board and inserting the code into the flash device. When the full circuit was built the software worked well on the Kanda ST7 Starter Kit and with the Flash circuit shown in Appendix 2.

ST7 Starter Kits are available from Kanda Systems Ltd. They are available as a 32pin version – Order code ST7KND1-Kit 2 Price $160 or 42pin or 56pin versions – Order code ST7KND2 – Kit 2 Price $160

Phone Kanda Sales Hotline +44 1970 621030 or Fax +44 1970 621040 e-mail sales@kanda.com or view the website www.kanda.com

Other auxillary parts to create the disco lights project are available from component suppliers like Maplin Electronics

Appendix 1

The code shown on the next page are the three files generated from the ST assembler, as mentioned earlier in this article. They include the set-up file, vector file, and the main program file:

Appendix 2

The schematic diagram shown in Figure 4 will enable the user to run the ST72254 flash device directly from the circuit that will connect to the power controller board via the 9-way 'D' type socket. The circuit will also allow the user to update the software to the device by simply inserting a keyfob. The keyfob is a device available for in-situ programming within the club environment, so eliminating the use of a PC for programming.

PERFECT MIRROR

Used as a Light Guide

by Reg Miles

Building on the 'perfect mirror' they created in 1998, Massachusetts Institute of Technology researchers have proposed a new kind of coaxial cable that may be able to carry light over long distances and around sharp bends while retaining its polarisation.

The 'perfect mirror' is so-dubbed because it combines the best of both metallic and dielectric mirrors. A conventional metallic mirror reflects light omnidirectionally; it also absorbs a significant portion of the incident light. A dielectric mirror is composed of multiple layers of transparent dielectric materials, which can be made to be extremely low loss compared to metal; and it reflects a prescribed range of frequencies coming from within a limited set of angles. The 'perfect mirror' reflects light from all angles and polarisations like a metallic mirror, but can be as low loss as a dielectric mirror, thanks to the properties of its multi-layer coating.

The researchers were Professors John D. Joannopoulos and Edwin L. Thomas, with Shanhui Fan and Yoel Fink, whose idea it was. Then, joined by Mihai Ibanescu, they put Fink's subsequent idea into practice and made a tube out of the 'perfect mirror' to create an omnidirectional waveguide. This has the significant advantage that it can bend the light in a shorter distance than an optical fibre; making it ideal for use in devices such as optical chips. It can also accomodate a greater bandwidth of light, with lower attenuation and less dispersion - thus getting more information to travel over longer distances.

Joannopoulos then proposed a coaxial version of it, with the inner and outer sheaths coated with the dielectric layers and the light propagated in the air-space between them. The researchers have launched a new company in Cambridge, Mass., called OmniGuide Communications, to explore the practicalities and possibilities of this and the multi-layer coatings.

According to the company: 'The OmniGuide Fibre could substantially reduce or even eliminate the need for amplifiers in optical networks. Secondly it will offer a bandwidth capacity that could potentially be several orders of magnitude greater than conventional single-mode optical fibres. Elimination of the need for optical amplifiers will cut the cost of deploying and maintaining optical networks. Combined with the increased bandwidth, it will allow network operators to slash the cost-per-bit dramatically.'

And according to Joannopoulos: 'What's important about this is that it has opened a new direction for experimental research that was not possible before. It's important to push along in this direction and see if we can find materials and fabrication approaches that will make this happen. We do know if we can do what the theory says, it will happen. This may be a breakthrough in bridging the very different requirements for transmitting infrared and radio frequencies at opposite ends of the energy spectrum. And the nice thing about it is that whatever you put in, you get out. This could make a big difference where polarisation is an issue.'

For additional information contact: **info@omni-guide.com**