PIC16F688 LED Stroboscoop ASM

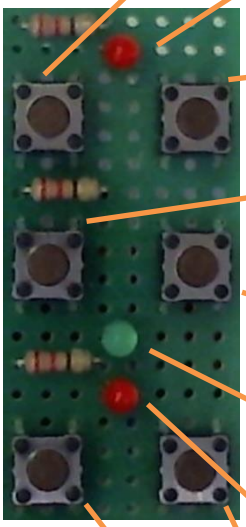
Beschrijving

Demo toestel om het stroboscopisch effect en toepassingen ervan te demonstreren.

Bediening

Na het ingeschakelen van de voeding wordt een test en opstartprocedure doorlopen. Alle ledjes knipperen en gaan na de test uit.

De stroboscoop staat nu automatisch in een vooraf ingestelde stand.



Verlaagd de knipper frequentie

Licht op wanneer een toets wordt ingedrukt.
Blijvend indrukken herhaald de toets functie.

Verhoogd de knipper frequentie

Verkleind de puls tijd en vergroot de pauze tijd.

Vergroot de puls tijd en verkleind de pauze tijd.

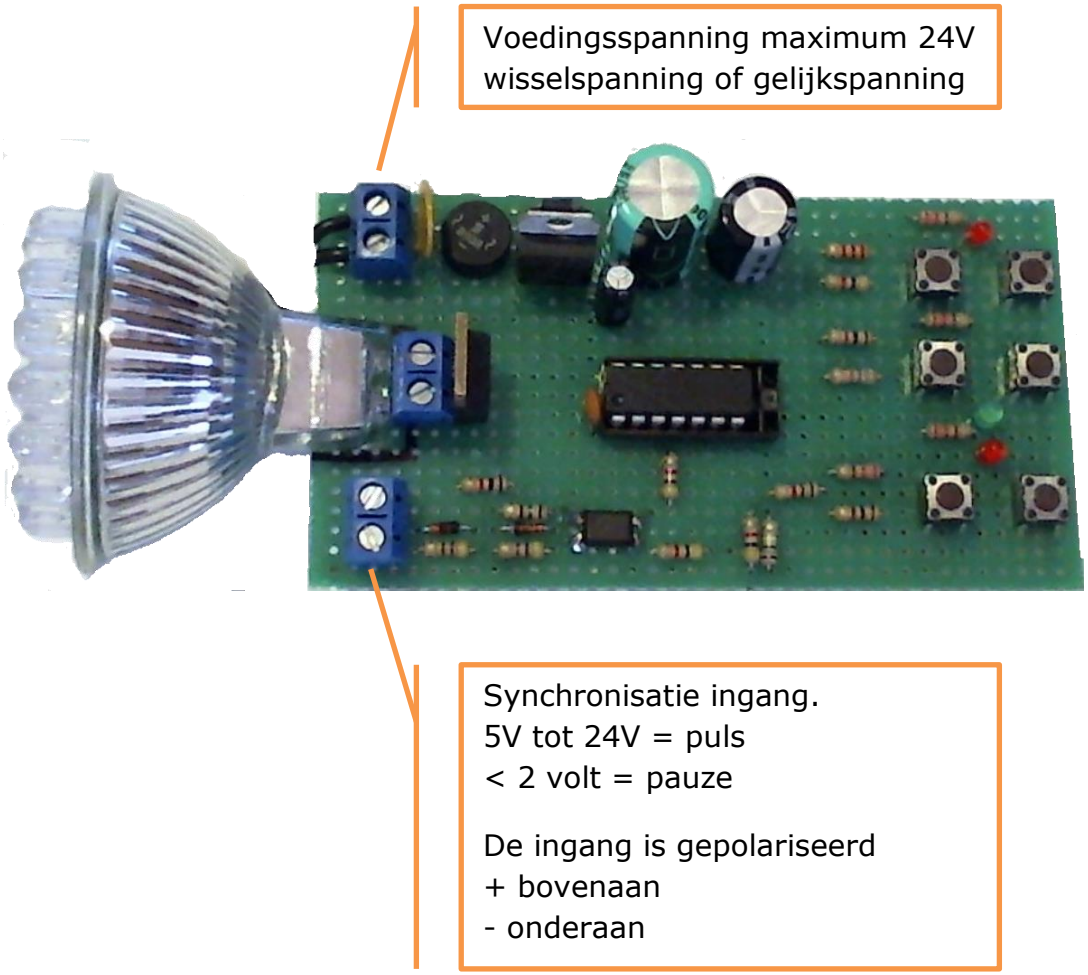
Brandt doorlopend bij manuele frequentie instelling.
Knippert in de “synchronisatie” mode.

Licht op wanneer in de “synchronisatie” mode een ingangspuls gedetecteerd wordt.

Selecteer de “synchronisatie” mode.

Selecteer de “manuele instelling” mode.

Aansluitingen



De maximum voedingsspanning is 24VDC of 24VAC .
De voedingsspanning ingang is niet gepolariseerd.
De klemmen van deze ingang zijn met “PWR” gemarkeerd.

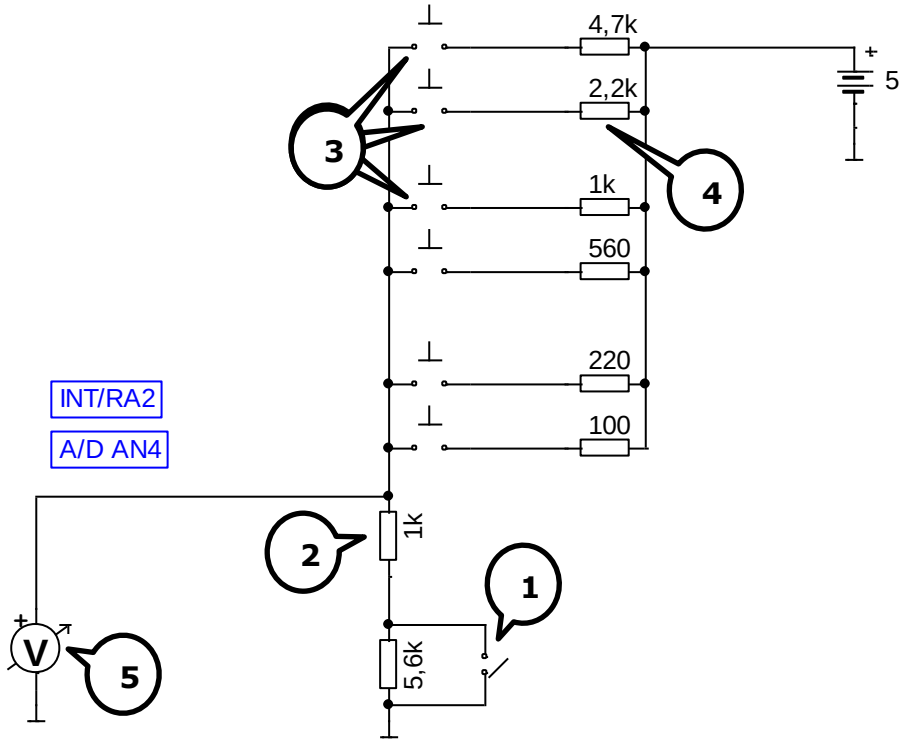
De synchronisatie ingang mag met elke spanning tot maximum 24V aangestuurd worden.
Deze ingang is gepolariseerd. Wanneer de polariteit van de aangelegde spanning niet overeen komt met deze die de schakeling verwacht wordt de synchronisatiepuls niet gedetecteerd.
Om een perfecte synchronisatie te bekomen moet het synchronisatiesignaal groter of grlijk zijn aan 5V voor een puls en kleiner dan 2V voor een pause.
De synchronisatie ingang vormt een belasting van minimum 1KΩ.

Details

Principiële werking van de bedieningstoetsen.

De schakeling wordt gevoed met 5V.
De uitgang van de schakeling (5) wordt digitaal geinterpreteerd met TTL niveaus waarbij laag maximaal 0.4V en hoog minimaal2.4V is .

De ingangsweerstand van de analoog digitaal converter parallel met de digitale ingang is ongeveer 47K.



- Stap 1

Alle schakelaars staan open de uitgangsspanning is bijgevolg 0V.

- Stap 2

De gebruiker drukt 1 of meerdere drukknoppen in (3).
De voedingsspanning wordt gedeeld over de serieweerstand van de drukknoppen en weerstand (2) 1K + 5K6 // Rin van de A/D // Rin van de digitale ingang.
In het slechtste geval wordt de 1ste drukknop met 4K7 serieweerstand ingedrukt.

$$Uuit = \frac{Uvoeding}{4700 + 5700} \times 5700$$

De uitgangsspanning wordt minimum 2.7V en triggert de digitale ingang.

- Stap 3

Met een digitale uitgang van de micro controller (MCU) wordt de weerstand van 5K6 kortgesloten. Schakelaar(1) is nu gesloten.

- Stap 4

De spanning over de A/D convertor(5) is nu.

$$Uuit = \frac{Uvoeding}{4700 + 1000} \times 1000$$

De analoog digitaal convertie wordt gestart.
De analoge waarde wordt door het programma volgens onderstaande tabel verwerkt.

AD CALCULATION						
VDD	5,00V					
R REF	1000R					
KEY	R VALUE	A/D Voltage	A/D 10bit	A/D 8bit	MIN	MAX
KEY1	4700R	0,88V	179	44	22	62
KEY2	2200R	1,56V	320	79	63	103
KEY3	1000R	2,50V	512	127	104	145
KEY4	560R	3,21V	656	163	146	186
KEY5	220R	4,10V	839	209	187	220
KEY6	100R	4,55V	930	231	221	243

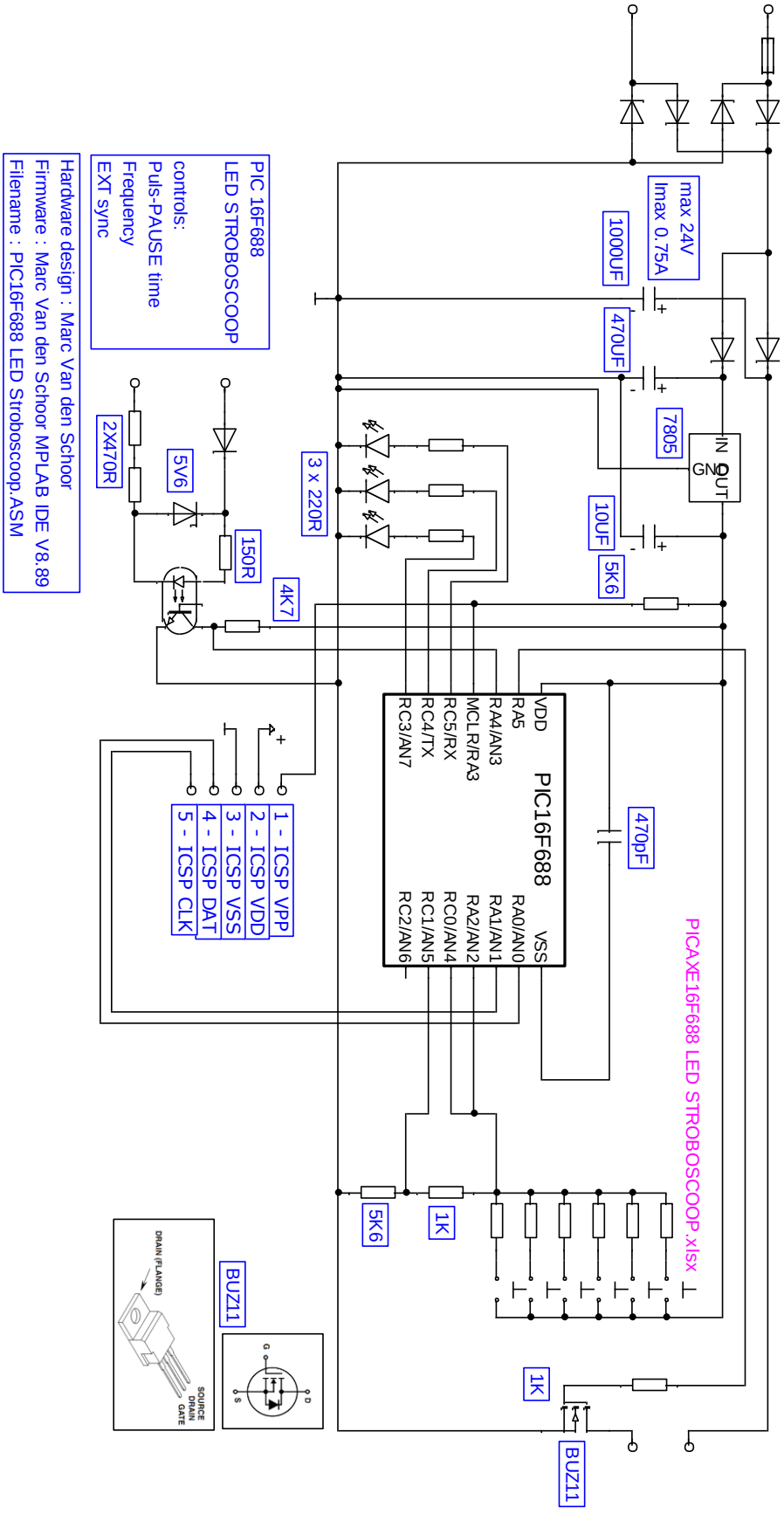
- Stap 5

Met een digitale uitgang van de micro controller (MCU) wordt de weerstand van 5K6 terug serie met de 1K weerstand(2) geschakeld.
Schakelaar(1) is nu open.

- Stap 6

Het programma in de MCU controlleerd of de toets al is losgelaten.
De schakeling is klaar voor de volgende toets detectie.

Schema


PIC 16F688
LED STROBOSCOOP
controls:
Puls-PAUSE time
Frequency
EXT sync
Hardware design : Marc Van den Schoor
Firmware : Marc Van den Schoor MPLAB IDE V8.89
Filename : PIC16F688 LED Stroboscoop.ASM
max 24V
Imax 0.75A
1000UF
470UF
7805
IN
GND
OUT
10UF
5K6
3 x 220R
2X470R
5V6
150R
4K7
PIC16F688
VDD
VSS
RA5
RA0/AN0
RA4/AN3
RA1/AN1
MCLR/RA3
RA2/AN2
RC5/RX
RC0/AN4
RC4/TX
RC1/AN5
RC3/AN7
RC2/AN6
1 - ICSP VPP
2 - ICSP VDD
3 - ICSP VSS
4 - ICSP DAT
5 - ICSP CLK
470pF
PICAXE16F688 LED STROBOSCOOP.xlsx
5K6
1K
1K
BUZ11
BUZ11
DRAIN (FLANGE)
SOURCE
DRAIN
GATE
G
D
S

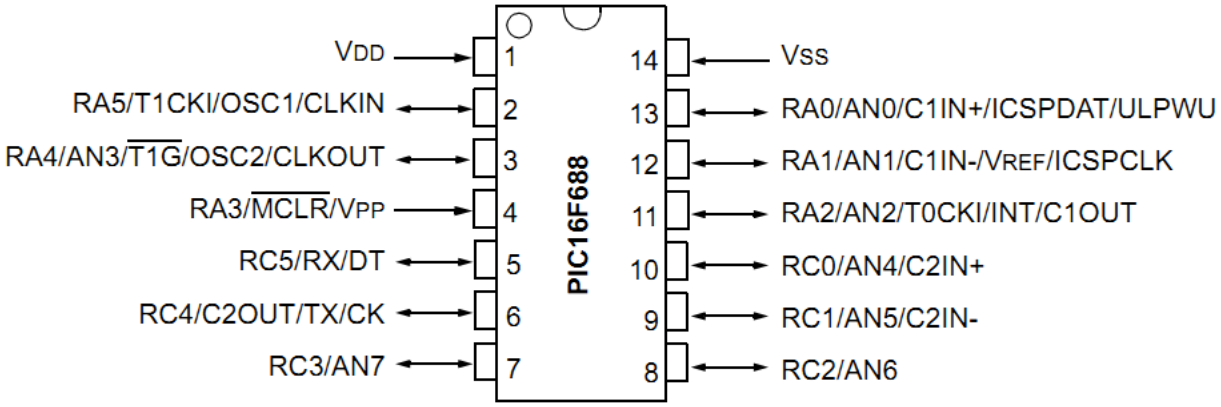


TABLE 12-2: PIC16F684 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	1, 2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1, 2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	–	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1, 2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1, 2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2, 3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2, 3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1, 2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1, 2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	–	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1, 2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1, 2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	1, 2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1, 2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1, 2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1, 2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	–	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	–	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	–	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	–	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$, $\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Assembler firmware

```
;*****************************************************************************
;                                                                            *
;   Filename:           PIC16F688 LED STroboscoop.asm                        *
;   Date:               02-2013                                              *
;   File Version:       release version 1                                    *
;                                                                            *
;   Author:             Marc Van den Schoor                                  *
;                                                                            *
;                                                                            *
;*****************************************************************************
;  editor setting                                                            *
;  fond                         Verdana 11                                   *
;                                                                            *
;*****************************************************************************
;                                                                            *
;   Files Required:     P16F688.INC                                          *
;                                                                            *
;*****************************************************************************
;                                                                            *
;   Notes:              ADRESH not supported in MPLAB SIM                    *
;                       INTERRUPT TIMER1 (8bit)                              *
;                       INTERRUPT RA2/INT                                    *
;                                                                            *
;                   LED RC3 : ON = manuel  -  flashing = EXT_sync    GREEN   *
;                   LED RC4 : flashing = Key Pressed                 RED     *
;                   LED RC5 : SYNC INPUT                             RED     *
;                                                                            *
;*****************************************************************************

        LIST            p=16F688        ; list directive to define processor
        include          "P16F688.inc"  ; processor specific variable definitions
        ERRORLEVEL      0,-302          ; suppress bank selection messages

;*****************************************************************
;MCU CONFIGURATION

        __CONFIG    _CP_OFF & _CPD_OFF & _BOD_OFF & _PWRTE_ON & _WDT_OFF &
_INTRC_OSC_NOCLKOUT & _MCLRE_ON & _FCMEN_OFF & _IESO_OFF

;*****************************************************************
;EEPROM DATA

        ORG     0x2100                  ; data EEPROM location
        DE      4D,56,64,53             ; MVdS copyright
```

```
;*****************************************************************
;VARIABLE DEFINITIONS

;general purpose registers BANK 0
                cblock  0x20                    ;start 0x20 end 0x7F
                ;ISR VARS
                        w_temp
                        status_temp
                        pclath_temp
                ;EEPROM DATA VARS
                        EEPROM_ADDR             ;EEPROM addres to read/write from to
                        EEPROM_DATA             ;EEPROM data
                ;DELAY VARS
                        Delay_VAR1              ;used in delay routine
                        Delay_VAR2              ;used in delay routine
                        Delay_VAR3              ;used in delay routine
                ;INPUT PROCESSING
                        ADC_VAR                 ;8 bit justified ADC value
                ;general purpose
                        X_VAR
                        Y_VAR
                        Z_VAR
                ;AD
                        AD_1                    ;8bit AD input value
                ;STROBOSCOOP
                        PULS_PAUSE_TIME
                        FREQUENCY
                        EXT_SYNC

                        FREQ                    ;EXT_SYNC mirror
                        PPT                     ;Pulse_pause_time mirror
                ;CTRL LED
                        LED_TIMER1
                        LED_TIMER2
                ;BUTTON KEY TRACKER
                        KEY_REPEAT
                endc

;general purpose registers BANK 1
                cblock          0xA0            ;start 0xA0 end 0xEF
                ;LOOPVAR
                        FSR1
                ;RESULT VAR
                        LOOKUP_VAR


                endc
```

```
;*****************************************************************
        ORG             0x000           ; processor reset vector
        goto            MAIN            ; go to beginning of program
;*****************************************************************
        ORG             0x004           ; interrupt vector location
        bcf             intcon,GIE      ; clear Global Interrupt Enable bit
                                        ;(disable all interrupts during ISR)
        movwf           w_temp          ; save off current W register contents
        movf            STATUS,w        ; move status register into W register
        movwf           status_temp     ; save off contents of STATUS register
        movf            PCLATH,w        ; move pclath register into W register
        movwf           pclath_temp     ; save off contents of PCLATH register

START_ISR_CODE      ; select interrupt source  ---------------------------------

        btfsc           INTCON,INTF     ;1 = The RA2/INT external interrupt occurred
        call            RA2_INTERRUPT   ;a key is pressed - process required action

        btfsc           INTCON,T0IF     ;1 = timer interrupt occured
        call            TIMER0_INT      ;timer 0 action

END_ISR_CODE        ; resume normal program flow  --------------------------

        movf            pclath_temp,w   ; retrieve copy of PCLATH register
        movwf           PCLATH          ; restore pre-isr PCLATH register contents
        movf            status_temp,w   ; retrieve copy of STATUS register
        movwf           STATUS          ; restore pre-isr STATUS register contents
        swapf           w_temp,f
        swapf           w_temp,w        ; restore pre-isr W register contents
        bsf             intcon,GIE      ; set Global Interrupt Enable bit

        retfie                          ; return from interrupt
```

```
;*****************************************************************
MAIN
                call            INIT
                call            LAMP_TEST       ;running light indicator LED's

                BSF             intcon,GIE      ;Start interrupt mode

                bsf             PORTC,RC4       ;GREEN LED ON = manuel frequency and
puls-pause timing

                ;call           LOOK_UP_FILL

                ;TEST PURPOSE: dissable timer 0 interrupt service routine
                ;bsf            intcon,T0IF     ; Timer0 Overflow Interrupt
                                                ; Flag bit must be cleared in software
                ;bcf            intcon,GIE      ; set Global Interrupt Enable bit
                ;bcf            intcon,T0IE     ; Timer0 Overflow Interrupt Enable bit

                ;TEST PURPOSE: check AD value by counting scope pulses
                ;call           AD_READ_TEST

RUN_PROG

                btfsc           KEY_REPEAT,0    ;SET = key repeat
                call            Delay100        ;if key repeat wait 100ms

                btfsc           KEY_REPEAT,0    ;SET = key repeat
                call            RA2_INTERRUPT   ;automatic key repeat

                call            EXTERNAL_SYNCHRONISATION

                goto            RUN_PROG

;*****************************************************************
RA2_INTERRUPT

                ; a key (button) is pressed
                ; set RC1 high before reading key AD value
                ; calculated A/D values in PICAXE16F688 LED STROBOSCOOP.xlsx

                BSF             PORTC,RC4       ;Push Button LED ON

                BANKSEL         TRISC           ;select bank1
                BCF             TRISC,RC1       ;0 = PORTC RC1 configured as an output
                BANKSEL         TMR0            ;select bank 0
                bcf             PORTC,RC1       ;short out 100K from the voltage divider
                call            AD_READ_1

                movlw           d'22'           ;AD value to low below 22
                subwf           AD_1,w          ;substract max value of key6 from AD value
```

```
                btfss       STATUS,C            ;if carry bit is set AD_1 AD
                                                ;is below alowable range
                goto        END_RA2_INTERRUPT

                movlw       d'62'               ;key1
                subwf       AD_1,w              ;substract max value of key6 from AD value
                btfss       STATUS,C            ;if carry bit is not set AD_1 > the max AD value of
key1
                goto        KEY_1

                movlw       d'103'              ;key2
                subwf       AD_1,w              ;substract max value of key6 from AD value
                btfss       STATUS,C            ;if carry bit is not set AD_1 > the max AD value of
key2
                goto        KEY_2

                movlw       d'145'              ;key3
                subwf       AD_1,w              ;substract max value of key6 from AD value
                btfss       STATUS,C            ;if carry bit is not set AD_1 > the max AD value of
key3
                goto        KEY_3

                movlw       d'186'              ;key4
                subwf       AD_1,w              ;substract max value of key6 from AD value
                btfss       STATUS,C            ;if carry bit is not set AD_1 > the max AD value of
key4
                goto        KEY_4

                movlw       d'220'              ;key5
                subwf       AD_1,w              ;substract max value of key6 from AD value
                btfss       STATUS,C            ;if carry bit is not set AD_1 > the max AD value of
key5
                goto        KEY_5

                movlw       d'243'              ;key6
                subwf       AD_1,w              ;substract max value of key6 from AD value
                btfss       STATUS,C            ;if carry bit is not set AD_1 > the max AD value of
key6
                goto        KEY_6

                goto        END_RA2_INTERRUPT   ;AD valu to high to be valid for key_6


KEY_1
                incf        FREQUENCY           ;increment FREQUENCY
                btfsc       STATUS,Z            ;if FREQUENCY > 255 then
                                                ;FREQUENCY = 255 FREQ = FREQ * 1ms
                decf        FREQUENCY
                movfw       FREQUENCY           ;on every freq change
                                                ;start with puls-pause as 1/2 frequency
                movwf       X_var
                RRF         X_VAR,f
```

```
                bcf     X_VAR,7
                movfw   X_VAR
                movwf   PULS_PAUSE_TIME
                goto    END_RA2_INTERRUPT


KEY_2
                decf    FREQUENCY               ;decrement FREQUENCY time
                btfsc   STATUS,Z                ;if FREQ time = 0 then FREQ = 1
                incf    FREQUENCY
                movfw   FREQUENCY               ;on every freq change
                                                ;start with puls-pause as 1/2freq
                movwf   X_var
                RRF     X_VAR,f
                bcf     X_VAR,7
                movfw   X_VAR
                movwf   PULS_PAUSE_TIME
                goto    END_RA2_INTERRUPT

KEY_3
                decf    PULS_PAUSE_TIME         ;decrement Pulse Pause time
                btfsc   STATUS,Z                ;if PP time = 0 then ON time = 1
                incf    PULS_PAUSE_TIME
                goto    END_RA2_INTERRUPT

KEY_4
                incf    PULS_PAUSE_TIME         ;increment Pulse Pause time
                btfsc   STATUS,Z                ;if PP time > 255 then ON time =
255
                decf    PULS_PAUSE_TIME
                MOVFW   FREQUENCY               ;PULS_PAUSE_TIME should
                                                ;never be higher than frequency
                SUBWF   PULS_PAUSE_TIME,W
                btfss   STATUS,C
                goto    END_RA2_INTERRUPT       ;PULS_PAUSE_TIME < = FREQ
                movfw   FREQUENCY               ;max PULS_PAUSE_TIME
                movwf   PULS_PAUSE_TIME         ;PULS_PAUSE_TIME >  FREQ
                                                ;THEN SET PPT = FREQ
                goto    END_RA2_INTERRUPT       ;OK

KEY_5
                bcf     EXT_SYNC,0              ;external synchronisation OFF
                bsf     intcon,T0IE             ;ENABLE Timer0 Interrupt
                bcf     INTCON,T0IF             ;clear Timer0 Overflow Interrupt
                                                ;Flag (req for ISR source select)
                BCF     PORTA,RA5               ;PULS RA5 OFF
                bsf     PORTC,RC4               ;GREEN LED ON = manuel
                                                ;freq and puls-pause timing
                goto    END_RA2_INTERRUPT
```

```
KEY_6

                bsf             EXT_SYNC,0                      ;external synchronisation ON
                bcf             intcon,T0IE                     ;DISABLE Timer0 Interrupt
                bcf             INTCON,T0IF                     ;clear Timer0 Overflow Interrupt
                                                                ;Flag (for cosmetic reason)
                BCF             PORTA,RA5                       ;PULS RA5 OFF
                bcf             PORTC,RC4                       ;GREEN LED OFF = manuel
                                                                ;freq and puls-pause timing
                movlw           d'255'
                movwf           LED_TIMER1
                movwf           LED_TIMER2
                goto            END_RA2_INTERRUPT


END_RA2_INTERRUPT


                BCF             KEY_REPEAT,0                    ;clear key repeat
                call            AD_READ_1
                movlw           d'22'                           ;lowest AD  value when any key
is pressed
                subwf           AD_1,w                          ;sub max value from AD value
                btfsc           STATUS,C                        ;
                BSF             KEY_REPEAT,0                    ;mark for key repeat


                ;SET RC1 as input to make the I/O tristate and add 100K to the voltage divider
                BANKSEL         TRISC                           ;select bank1
                BSF             TRISC,RC1                       ;1 = PORTC pin configured as an
input (tri-stated)
                BANKSEL         TMR0                            ;select bank 0
                ;BCF            PORTC,RC1                       ; TRISTATE add 100K to the
voltage divider

                BCF                     PORTC,RC3               ;Push Button LED OFF
                bcf                     INTCON,INTF             ;RA2/INT external interrupt
                                                                ;(must be cleared in software)
                retlw           0x0
```

```
;*****************************************************************
TIMER0_INT

                btfsc       EXT_SYNC,0          ;external synchronisation OFF = 0
                goto        TIMER0_INT_END      ;external synchronisation ON = 1


                movfw       PPT                 ;PPT already 0?
                btfsc       status,z
                BCF         PORTA,RA5           ;PULS RA5 OFF
                movfw       PPT                 ;PPT already 0?
                btfsc       status,z
                goto        PPT_PAUSE
                decf        PPT
PPT_PAUSE

                movfw       FREQ                ;freq 0 ?
                btfsc       STATUS,Z
                goto        FREQ_0

                decf        FREQ                ;count douwn at a rate of 0.512ms

                goto        TIMER0_INT_END

FREQ_0
                MOVFW       FREQUENCY           ;end of cycle - RELOAD
                MOVWF       FREQ
                MOVFW       PULS_PAUSE_TIME
                MOVWF       PPT
                BSF         PORTA,RA5           ;PULS RA5 ON



TIMER0_INT_END
                bcf         INTCON,T0IF         ;Timer0 Overflow Interrupt Flag bit
                                                ;must be cleared in software
                retlw       0x0
```

```
;*****************************************************************
EXTERNAL_SYNCHRONISATION

        ;sample rate > 100KHz
        btfss       EXT_SYNC,0                      ;external synchronisation ON = 1
        goto        EXTERNAL_SYNCHRONISATION_END    ;external synchronisation OFF = 0

        btfsc       PORTA,RA4                       ;pol input = NOT(real input)
        goto        NO_EXT_INPUT
        goto        EX_INPUT

NO_EXT_INPUT
        BCF         PORTA,RA5                       ;INPUT to folow is LOW
        BCF         PORTC,RC5                       ;INPUT to folow is LOW
        goto        END_EXT_INP_TEST

EX_INPUT
        BSF         PORTA,RA5                       ;INPUT to folow is HIGH
        BSF         PORTC,RC5                       ;INPUT to folow is HIGH
END_EXT_INP_TEST

        DECFSZ      LED_TIMER1                      ;GREEN LED blinker stage 1
        goto        EXTERNAL_SYNCHRONISATION_END
        movlw       d'200'
        movwf       LED_TIMER1
        DECFSZ      LED_TIMER2                      ;GREEN LED blinker stage 2
        goto        EXTERNAL_SYNCHRONISATION_END
        btfss       PORTC,RC4
        goto        BLINK_ON
        goto        BLINK_OFF
BLINK_ON
        BSF         PORTC,RC4
        goto        BLINK_END
BLINK_OFF
        BCF         PORTC,RC4                       ;blink GREEN LED
BLINK_END
        MOVLW       d'100'
        MOVWF       LED_TIMER2

EXTERNAL_SYNCHRONISATION_END

        Return
```

```
;*****************************************************************
LOOK_UP_FILL                                ;load lookup table with sequential data in
rambank1

                banksel     ADCON1          ;bank1
                movlw       0xA3
                movwf       FSR             ;preload with first free GPR adres in rambank1
                clrf        FSR1            ;clear indirect file acces counter
LOOK_UP_1

                incf        FSR1
                movfw       FSR1
                movwf       LOOKUP_VAR
                movwf       INDF

                incf        FSR
                movlw       0xF0            ;end adres is EF. This is the highest adres that
can be written to
                subwf       FSR,w           ;substract end adres. When result is 0 then FSR
at end adres
                btfss       status,Z        ;check zero flag, if zero then done
                goto        LOOK_UP_1

                banksel     ADCON0          ;bank0

LOOK_UP_READ

                banksel     ADCON1          ;bank1
                movlw       0xA3
                movwf       FSR             ;preload with first free GPR adres in rambank1
                clrf        FSR1            ;clear indirect file acces counter
                clrf        LOOKUP_VAR      ;clear lookup var

LOOK_UP_2

                movfw       INDF
                movwf       LOOKUP_VAR

                incf        FSR
                movlw       0xF0            ;end adres is EF. This is the highest adres that
can be written to
                subwf       FSR,w           ;substract end adres. When result is 0 then FSR
at end adres
                btfss       status,Z        ;check zero flag, if zero then done
                goto        LOOK_UP_2

                banksel     ADCON0          ;bank0



                retlw       0x0
```

```
;*****************************************************************
;check A/D value as puls train on bitscope hardware
;only usedduring development
;output port set as requered

AD_READ_TEST


                btfsc           AD_1,7
                goto            BIT7_SET

                btfsc           AD_1,6
                goto            BIT6_SET

                btfsc           AD_1,5
                goto            BIT5_SET

                btfsc           AD_1,4
                goto            BIT4_SET

                btfsc           AD_1,3
                goto            BIT3_SET

                btfsc           AD_1,2
                goto            BIT2_SET

                btfsc           AD_1,1
                goto            BIT1_SET

                btfsc           AD_1,0
                goto            BIT0_SET

                goto            BIT_NOT_SET

BIT7_SET
                call            delay5
                movlw           d'8'
                movwf           Y_VAR
BIT7_LOOP
                bsf             portc,RC5
                call            delay1
                bcf             portc,RC5
                call            delay1
                decfsz          Y_VAR
                goto            BIT7_LOOP
                goto            BIT_NOT_SET
```

```
BIT6_SET
                call            delay5
                movlw           d'7'
                movwf           Y_VAR
BIT6_LOOP
                bsf             portc,RC5
                call            delay1
                bcf             portc,RC5
                call            delay1
                decfsz          Y_VAR
                goto            BIT6_LOOP
                goto            BIT_NOT_SET


BIT5_SET
                call            delay5
                movlw           d'6'
                movwf           Y_VAR
BIT5_LOOP
                bsf             portc,RC5
                call            delay1
                bcf             portc,RC5
                call            delay1
                decfsz          Y_VAR
                goto            BIT5_LOOP
                goto            BIT_NOT_SET

BIT4_SET
                call            delay5
                movlw           d'5'
                movwf           Y_VAR
BIT4_LOOP
                bsf             portc,RC5
                call            delay1
                bcf             portc,RC5
                call            delay1
                decfsz          Y_VAR
                goto            BIT4_LOOP
                goto            BIT_NOT_SET

BIT3_SET
                call            delay5
                movlw           d'4'
                movwf           Y_VAR
BIT3_LOOP
                bsf             portc,RC5
                call            delay1
                bcf             portc,RC5
                call            delay1
                decfsz          Y_VAR
                goto            BIT3_LOOP
                goto            BIT_NOT_SET
```

```
BIT2_SET
                call            delay5
                movlw           d'3'
                movwf           Y_VAR
BIT2_LOOP
                bsf             portc,RC5
                call            delay1
                bcf             portc,RC5
                call            delay1
                decfsz          Y_VAR
                goto            BIT2_LOOP
                goto            BIT_NOT_SET

BIT1_SET
                call            delay5
                movlw           d'2'
                movwf           Y_VAR
BIT1_LOOP
                bsf             portc,RC5
                call            delay1
                bcf             portc,RC5
                call            delay1
                decfsz          Y_VAR
                goto            BIT1_LOOP
                goto            BIT_NOT_SET

BIT0_SET
                call            delay5
                movlw           d'1'
                movwf           Y_VAR
BIT0_LOOP
                bsf             portc,RC5
                call            delay1
                bcf             portc,RC5
                call            delay1
                decfsz          Y_VAR
                goto            BIT0_LOOP
                goto            BIT_NOT_SET


BIT_NOT_SET
                retlw           0x0
```

```
;*****************************************************************

AD_READ_1                               ;read ADC

        bcf     portc,RC1               ;set RC3 high to short-out R that generates
                                        ;5V on int/RA2

        MOVLW   b'00010001'             ;Left justify (read 8MSB from ADFM)
        MOVWF   ADCON0                  ;Vdd Vref, AN4 = 100, AN5 = 101 / AD stopped
        BSF     ADCON0,GO               ;Start conversion
        CALL    Delay5                  ;Acquisiton delay (calculated required 48uS)

        BCF     ADCON0,GO
        BTFSC   ADCON0,GO               ;Is conversion done?
        GOTO    $-1                     ;No, test again

        ;BANKSEL ADRESH                 ;conversion completed
        MOVF    ADRESH,W                ;Read 8bit AD
        MOVWF   AD_1                    ;store

        bsf     portc,RC3               ;set RC3 LOW

        retlw   0x00
```

```
;*****************************************************************
 ;DELAY ROUTINES (8 MHz clock)
Delay255
                movlw       d'255'              ;delay 255mS
                goto        d0

Delay100
                movlw       d'100'              ;delay 100mS
                goto        d0

Delay50
                movlw       d'50'               ;delay 50mS
                goto        d0

Delay5
                movlw       d'5'                ;delay 5ms
                goto        d0

Delay1
                movlw       0x01                ;delay 1ms
d0
                movwf       Delay_VAR1
d1
                movlw       0x90                ;timing standard delay = 1mS
                movwf       Delay_VAR2
                movlw       0x02
                movwf       Delay_VAR3
Delay_0
                decfsz      Delay_VAR2, f
                goto        $+2
                decfsz      Delay_VAR3, f
                goto        Delay_0

                decfsz      Delay_VAR1,f
                goto        d1

                retlw       0x00
```

```
;*****************************************************************
LAMP_TEST

                movlw       d'5'
                movwf       X_var

LAMP_TEST_LOOP

                bsf         PORTC,RC3               ;
                call        Delay50
                bcf         PORTC,RC3               ;
                bsf         PORTC,RC4               ;
                call        Delay50
                bcf         PORTC,RC4               ;
                bsf         PORTC,RC5               ;
                call        Delay50
                bcf         PORTC,RC5               ;
                call        Delay50

                decfsz      X_var
                goto        LAMP_TEST_LOOP

                return
```

```
;*****************************************************************
INIT

        banksel     OSCCON
        CLRF        OSCCON
        MOVLW       b'01110111'
        movwf       OSCCON

;I/O SETUP

;PORT A
        BANKSEL     PORTA                   ;
        CLRF        PORTA                   ;Init PORTA
        BANKSEL     ANSEL                   ;
        CLRF        ANSEL                   ;NO analog I/O
                                            ;RA0,RA1,RA2,RA3,RA4,RA5 digital I/O
        MOVLW       b'00011111'             ;RA0,RA1,RA2,RA3,RA4 = inputs
                                            ;RA5 = output
        MOVWF       TRISA                   ;

;PORT C
        BANKSEL     PORTC                   ;
        CLRF        PORTC                   ;Init PORTC
        CLRF        CMCON0                  ;disable comparator function
        BANKSEL     ANSEL                   ;
        MOVLW       b'00010000'             ;RC0/AN4 analog input
                                            ;RC1, RC2, RC3, RC4, RC5 digital I/O
        MOVWF       ANSEL                   ;
        MOVLW       b'00000111'             ;RC0,RC1,RC2 = inputs
                                            ;RC1,RC3,RC4,RC5 = output
        MOVWF       TRISC                   ;


;CONFIGURE ANALOG CHANNELS ON PORT C
        BANKSEL     ADCON1                  ;
        MOVLW       b'00100000'             ;ADC ATD FOSC/32clock
        MOVWF       ADCON1                  ;
        BANKSEL     ADCON0                  ;000xxx00 set for chn select (AN4=100)
        MOVLW       b'00010000'             ;Left justify (read 8MSB from ADFM)
        MOVWF       ADCON0                  ;Vdd Vref, AN4 = 100 / AD stopped
```

```
;TIMER 0 interrupt
        BANKSEL     OPTION_REG              ;corrected options for this prog  b'01000XXX'
        MOVLW       b'01000010'             ;Prescaler Rate Select bits b'-----XXX' devide
(fosc/4)/xxx
        MOVWF       OPTION_REG              ;
        bcf         intcon,T0IF             ;Timer0 Overflow Interrupt
                                            ;Flag bit must be cleared in software
        bcf         intcon,GIE              ;Global Interrupt Enable bit = 0
                                            ;DISABLE INTERRUPT.
                                            ;Start interrupt after init is completed
        bsf         intcon,T0IE             ;Timer0 Overflow Interrupt Enable bit

        BANKSEL     TMR0                    ;make sure bank 0 is selected

;INT/RA2 interrupt
        BANKSEL     OPTION_REG
        bsf         OPTION_REG,INTEDG       ;Interrupt on rising edge of RA2/INT pin
        BANKSEL     INTCON
        bcf         INTCON,INTF             ;1 = The RA2/INT  clear to enable selection
        bsf         INTCON,INTE             ;1 = Enables the RA2/INT external interrupt
        BANKSEL     porta                   ;make sure bank 0 is selected

;VARIABLE PRESET/RESET

        CLRF        ADC_VAR
        CLRF        X_VAR
        CLRF        Y_VAR
        CLRF        Z_VAR
        CLRF        TEMP
        CLRF        EXT_SYNC                ;external synchronisation OFF default
        CLRF        KEY_REPEAT              ;key repeat OFF
        MOVLW       0X1
        MOVWF       LED_TIMER1
        MOVLW       0X1
        MOVWF       LED_TIMER2
        MOVLW       d'40'
        MOVWF       FREQUENCY
        MOVLW       d'10'
        MOVWF       PULS_PAUSE_TIME
        bcf         EXT_SYNC,0              ;default to manuel control (external sync OFF)
                                            ;SET RC1 as input to make the I/O tristate and
                                            ;add 100K to the voltage divider
        BANKSEL     TRISC                   ;select bank1
        BSF         TRISC,RC1               ;1 = PORTC pin configured as an input (tri-stated)
        BANKSEL     TMR0                    ;select bank 0
        CLRF        PORTC

        retlw   0x0                                 ;init is ready

;*****************************************************************
        END             ; end of program
```