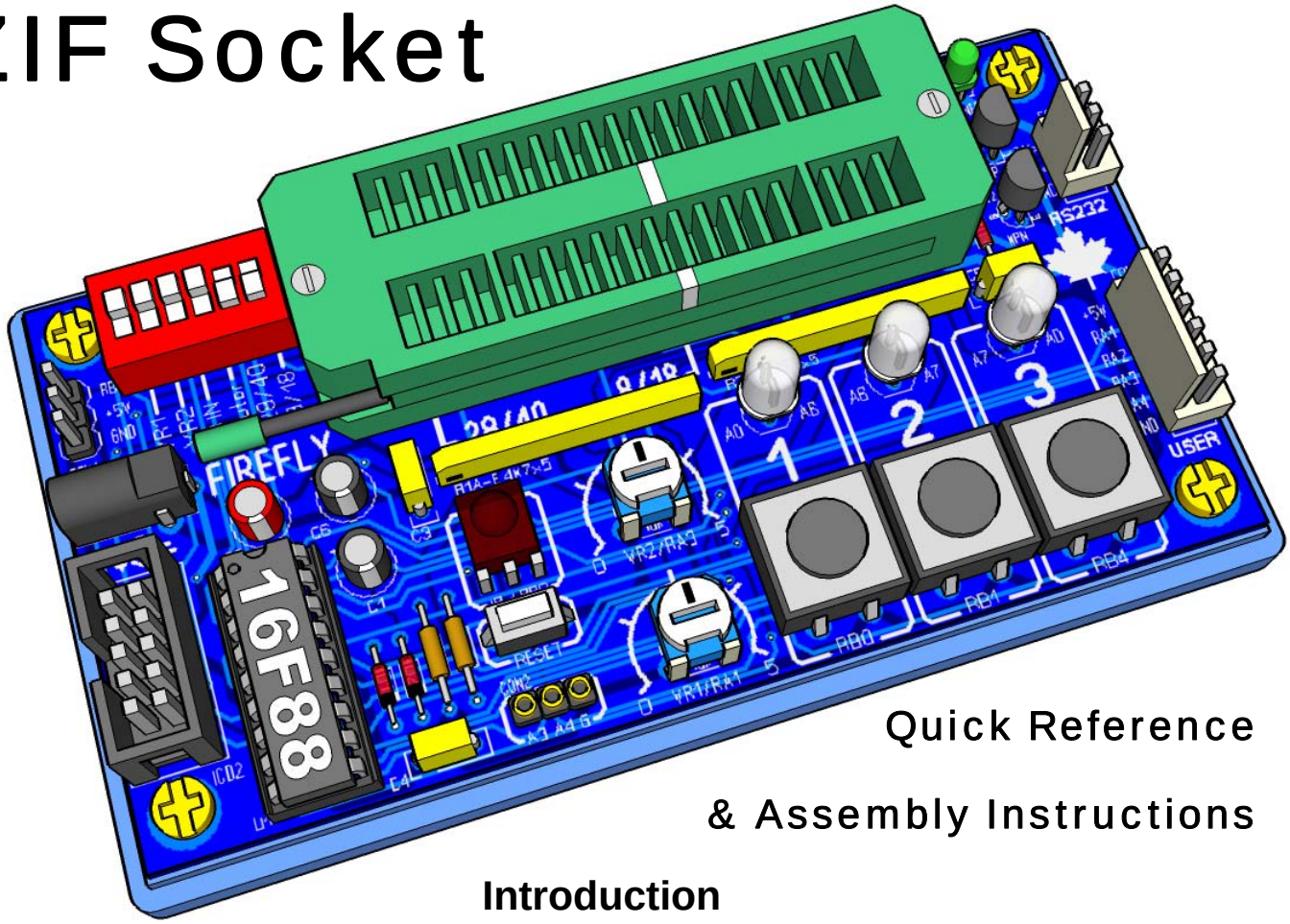
FIREFLY 16F88 Tutor & ZIF Socket



Quick Reference

& Assembly Instructions

Introduction

It's surprising how many people wanted to get into microcontroller programming but have no idea what to do for their first project let alone how to even program one. Secondly even though ICP is available there is still a demand for ZIF (Zero Insertion Force) programming sockets. So I combined two kits into one. Firefly can be built as a 16F88 trainer, a PIC ZIF programming socket or both. The Firefly was designed as a PIC tutor companion kit to the Inchworm ICD2.

16F88 tutor highlights

- Microchip PIC16F88 with programmable internal 8 speed oscillator, 31.25 KHz to 8MHz
- 3 dual color Red / Green LEDs & 3 User Pushbuttons (two sizes available economy / large)
- 38KHz Infrared sensor & 2 Switch Selectable Variable Resistors for 0-5V on RA1 / RA3
- RS232 connector with inverting transistors *(requires optional DE9 to 3pin cable adapter)*
- 6 pin User I/O port for external expansion and your own projects
- 3 pin servo style connector for Servos, IR LED, Buzzer, Piezo speakers, switches etc.
- 3 hole iButton® / 1-wire and quick test socket, small 57mm x 107mm PCB
- ZIF socket for programming 8/14/18/20*/28/40 pin Flash PICs
- Mode switch, Reset switch, 1.3mm 5VDC power jack & Inchworm ICD2 connector

**although not on the silk-screening 20pin devices can be programmed using the 8/18 setting*

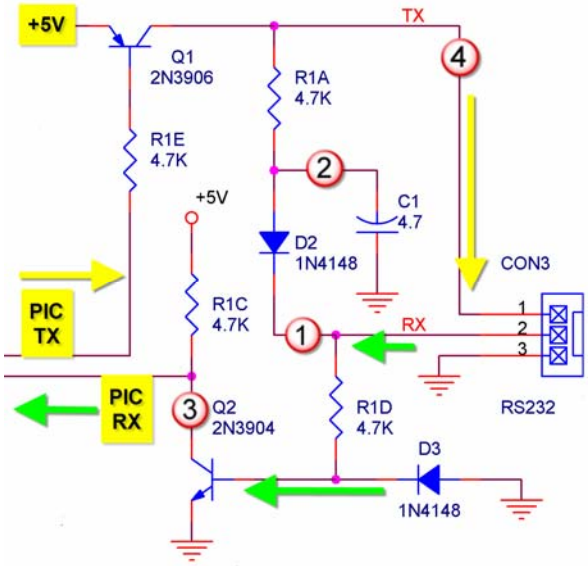
Why the PIC16F88…

	16F84A	16F628A	16F88
Flash	1024	2048	4096
RAM	68	224	368
EEPROM	64	128	256
Internal Osc	No	2 speed	8 speed
A/D	None	None	7 @ 10bit
Timers	1	3	3
USART	No	Yes	Yes
PWM / CCP	No	1	1
Comparators	No	2	2
Self Flashing	No	No	Yes
Debugger	No	No	Yes

The PIC16F88 is the current king of the 18pin 16F series PICs. This inexpensive and very powerful micro is just packed with internal peripherals. Although there are literally hundreds of PICs to choose from the 16F88 stood on top of the pack for one reason more than any other; **the 16F88 has the hardware debugger built in.** Whether you're new with PICs or old school, when you're working with any microcontroller there will be a time when you wish you could see what that little black box was thinking on the inside. Well that's what a debugger lets you do; you can set breakpoints *stop a running program* and see exactly what the processor sees. You can even modify the data. Difficult to find program errors can be spotted and wiped out with relative ease and as a learning tool the mysteries of the internal workings are quickly revealed.

Circuit Description

A detailed description of Firefly built in peripherals can be found in the "Firefly User Guide" Although Firefly is a fairly straight forward design; one feature worth further examination would be the RS232 level shifter. When using the hardware USART the data on the RX & TX pins has to be inverted and level shifted to be useful as RS232. This type of RS232 inverter requires the hosts idle *"Mark"* TX voltage is at or below GND *(ideally lower than -3V).*



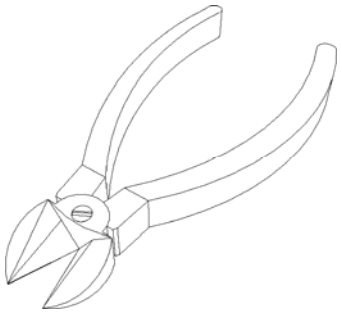
1. D2 borrows power from the target RS232 TX pin.
2. C1 charges to a negative voltage when RX is idling and R1A pulls TX to this negative (idle / mark) voltage.
3. Inverted *by Q2* (RX incoming RB2) from target RS232.
4. Inverted *by Q1* (TX outgoing RB5) from U1-16F88.

**If your target serial port does not pull RX to a negative voltage when idle; the serial port may not operate because the TX line will never pull to a MARK (negative) voltage. Test the voltage at point* ①

Other noteworthy features would be the multiplexed (Mux) LED display. The Red / Green two lead LEDs can only be turned on one at a time. This scheme; sometimes called Charlieplexing allows very few I/O pins to run many LEDs. With the Firefly we can effectively control six LEDs with three pins, to make more than one LED appear lit at a time; but to do this we have to continuously refresh the display faster than 40Hz. This is not difficult to program, as an example try removing all the "call Delay" lines in Blinky 16F88 on page 9 and the result will be the LEDs will appear Orange *(they're actually only lit one at a time, try single stepping in debug mode and you'll see).*

The IR1 decoder TSOP34838 has an internal 38KHz band pass filter so it will only see IR transmissions that are modulated at or near 38KHz; this is designed to prevent ambient light from falsely triggering the sensor. IR1 also shares RB0 with Pushbutton SW2. Therefore when you're using SW2 *Pushbutton 1* you should disable IR1 using SW5-3 ⌨ the reverse is true when using IR1 ⌨ (don't use SW2)

Necessary Tools *(not included in kit)*

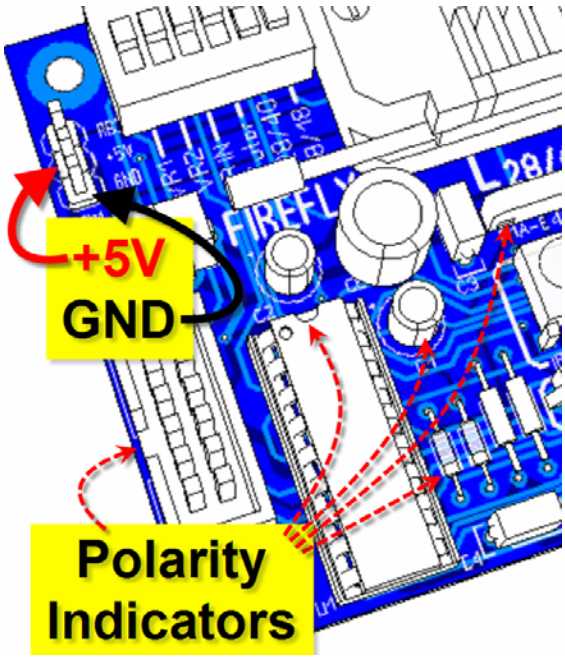


As with any electronic kit the following tools are essential:

- Low wattage fine tip soldering iron <50W
- Resin core solder
- Wire cutters or side cutters (small)
- Needle nose pliers (small)
- Slotted screwdriver (small)
- Phillips screwdriver (small)
- Multimeter (this really is a must for any electronics project)

Assembly

Traditionally it's easiest to assemble a circuit board with the lowest profile and/or smallest parts first.



Install 5% resistors R2, R4, R5
Small signal diodes D1 thru D3 (Red 1N4148)
Note: diodes use a colored band to denote polarity
SIP resistors *Isolated type* R1 (4.7K) ,R3 (100ohm)
(You can use 1/8W resistors if you don't have the SIPs)
IC socket for U1 *(notice notch orientation)*
Transistors Q1 (2N3906) / Q2 (2N3904)
Power LED4 (green)
LED1, LED2, LED3 (Dual Color RED / GREEN)
Note: LEDs have a flat side indicating polarity
Capacitors C3, C4, C5 *(note lead spacing)*
Capacitors C1, C6 (4.7uF)
Capacitor *(note polarity)* C2 (33uF)
Switches SW1, SW5 *(note pin 1 on SW5)*
Switches SW2, SW3, SW4 *(one of two sizes will fit)*
38KHz IR decoder IR1 *(see the cover diagram for details)*
Optional 1.3mm coax power jack P1 *regulated 5V required*
Connectors CON1 thru CON5
(Use the Firefly cover illustration as a guide)
Optional install Aries ZIF socket. *(The 3M Textool ZIF will not fit and is not recommended)*
Note: The Aries ZIF socket and SW5 can be a very tight fit. It's advised to place them both on the board before soldering either.

Initial Testing

Before installing IC1 (PIC16F88) apply power to the board, this is best done by attaching an ICD2 such as the Inchworm to the ICD2 connector (CON5). The 3mm green power LED4 should be on. *Note: The #2 LED may be on (Green) when the 16F88 is blank.* This is normal as the comparators (which by defalt are on) are driving the pins.

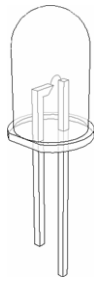
Final Assembly

Make sure the board is un-powered then install U1 (PIC16F88). If you've built a Tutor try out the program "Blinky VR1" near the end of this document for testing the BiColor LEDs and VR1.
There will be Firefly and other PIC related projects added to the blueroomelectronics site regularly. See the project link on the site for details.

Make sure to download the 16F88 datasheet from Microchip for a detailed explanation of the 16F88.

Parts List FIREFLY T (Tutor)

Capacitors

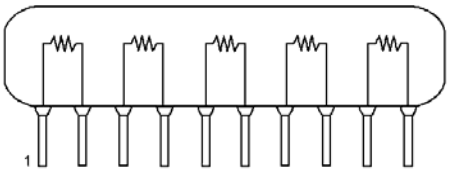


1	C1	4.7uF 16V (subminiature)
1	C2	33uF 6.3V (subminiature 22uf thru 47uf may be used)
3	C3,4,5	0.1uF
1	C6	4.7uF 6.3V (subminiature 4.7uf thru 47uf may be used)

Resistors ¼W (tan body, 4 color bands)

1	R2	22K	Red, Red, Orange, Gold
2	R4,5	330	Orange, Orange, Brown, Gold
1	R1	4.7K x 5	SIP 10pin 5 independent resistors
1	R3	100 x 5	SIP 10pin 5 independent resistors
2	VR1,VR2	10K	10K Trimmer potentiometers

Semiconductors

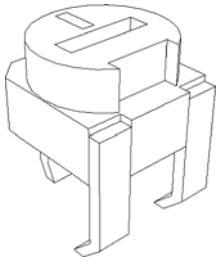


3	D1,2,3	1N4148 Small Signal Diode
3	LED1,2,3	3mm or 5mm RED/GREEN dual color LED (LTL-293SJW 2 lead)
1	LED4	3mm GREEN LED
1	IR1	TSOP34838 38KHz IR detector
1	Q1	2N3906 PNP (EBC)
1	Q2	2N3904 NPN (EBC)
1	U1	PIC16F88

Switches

1	SW1	Small pushbutton (450-1173-ND)
3	SW2,3,4	Pushbutton (Small or Large 450-1131-ND see text)
1	SW5	DIP Switch 6 position

Connectors



1	CON1	6pin Molex connector	+5V,RA1,RA2,RA3,RA4,GND
1	CON2	3pin machine socket	RA3,RA4,GND
1	CON3	3pin Molex connector	TX,RX,GND
1	CON4	3pin header	RB3,+5V,GND
1	CON5	ICD 2x5 PCB Male	ICD2 (Inchworm)

Miscellaneous

1	18-pin	IC Socket
1	P1	1.3mm PCB RA Coax Jack (optional)

Optional Accessories

1	Enclosure	Hammond 1591B (112mm x 62mm) or 1591BC clear lid
1	5VDC Adapter	5VDC 1.3mm center positive coax AC adapter
1	3pin RS232 Cable	See text for building instructions

Parts required for Firefly Z (ZIF) option*

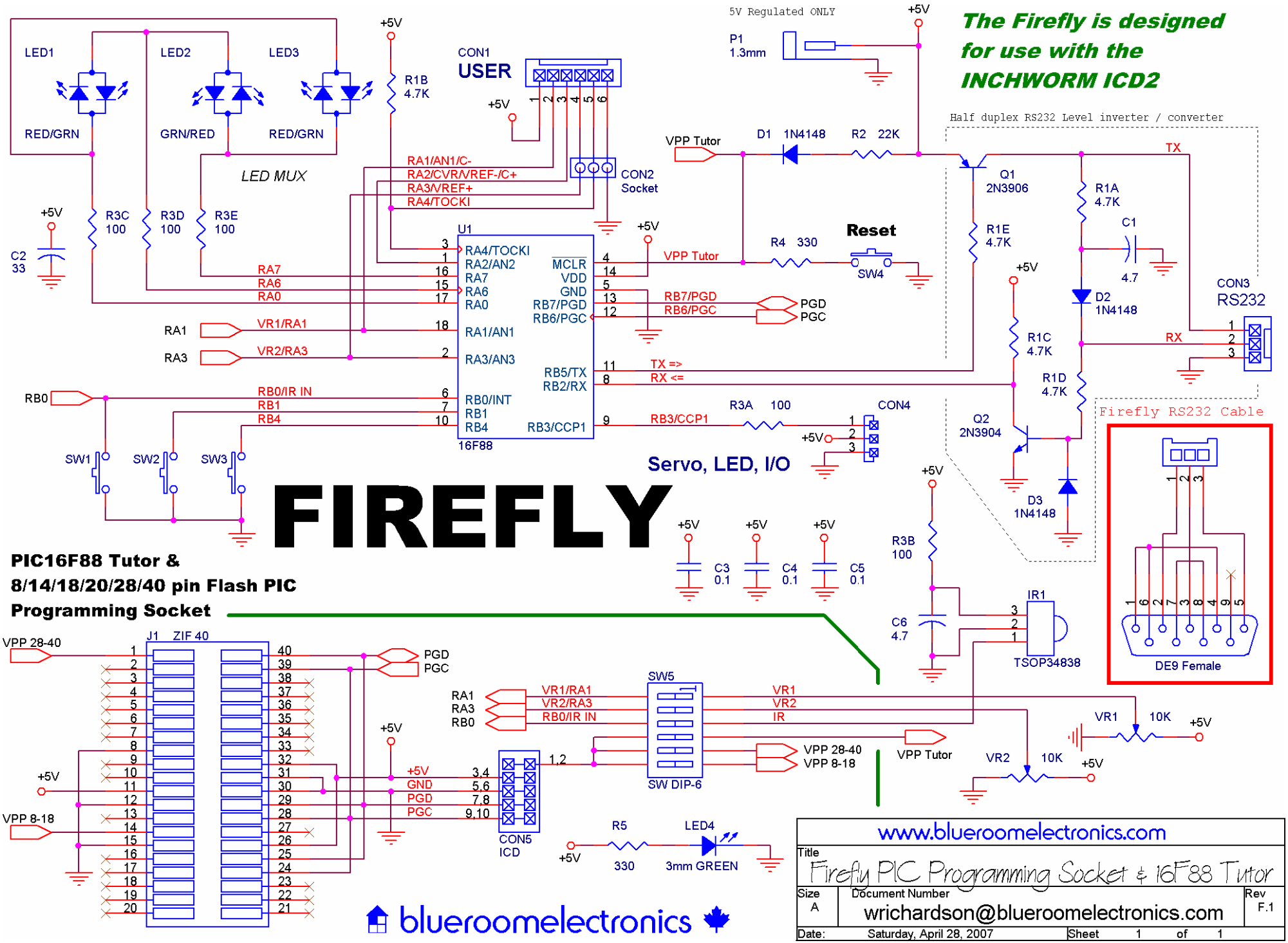
1	J1	Aries 40-6554-10 (*3M / Textool may require modification*)

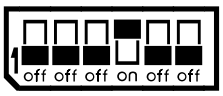
**To build a "Firefly Z" ZIF only option only J1, SW5, CON5 & C3 are required. LED4 and R5 are optional but recommended.*

Parts required for Firefly TZ (Tutor & ZIF)

The Firefly TZ requires all parts listed above

Part numbers ending in -ND can be found at http://www.Digikey.com


The Firefly is designed for use with the INCHWORM ICD2
FIREFLY
Servo, LED, I/O
PIC16F88 Tutor &
8/14/18/20/28/40 pin Flash PIC
Programming Socket
LED1
LED2
LED3
RED/GRN
GRN/RED
RED/GRN
LED MUX
R1B 4.7K
R3C 100
R3D 100
R3E 100
C2 33
CON1
USER
CON2
Socket
5V Regulated ONLY
P1
1.3mm
Half duplex RS232 Level inverter / converter
VPP Tutor
D1 1N4148
R2 22K
Q1 2N3906
R1A 4.7K
R1E 4.7K
C1
4.7
D2 1N4148
R1C 4.7K
R1D 4.7K
Q2 2N3904
D3 1N4148
CON3
RS232
TX
RX
Reset
SW4
R4 330
U1
16F88
RA4/TOCKI
RA2/AN2
RA7
RA6
RA0
RA1/AN1
RA3/AN3
RB0/INT
RB1
RB4
MCLR
VDD
GND
RB7/PGD
RB6/PGC
RB5/TX
RB2/RX
RB3/CCP1
RA1/AN1/C-
RA2/CVR/VREF-/C+
RA3/VREF+
RA4/TOCKI
VR1/RA1
VR2/RA3
RB0/IR IN
RB7/PGD
RB6/PGC
TX =>
RX <=
PGD
PGC
SW1
SW2
SW3
R3A 100
CON4
C3 0.1
C4 0.1
C5 0.1
R3B 100
C6 4.7
IR1
TSOP34838
Firefly RS232 Cable
DE9 Female
J1 ZIF 40
VPP 28-40
VPP 8-18
SW5
SW DIP-6
VR1
VR2
IR
VPP 28-40
VPP 8-18
VPP Tutor
VR1 10K
VR2 10K
CON5
ICD
R5 330
LED4
3mm GREEN
blueroomelectronics
www.blueroomelectronics.com
Title
Firefly PIC Programming Socket & 16F88 Tutor
Size A
Document Number
wrichardson@blueroomelectronics.com
Rev F.1
Date: Saturday, April 28, 2007
Sheet 1 of 1



DIP Switch functions (SW5)

In the upper left corner of the Firefly there is a six position DIP switch (shown in Tutor default mode)

Functions from 1-6 (left to right) *enable = on / up & disable = off / down*

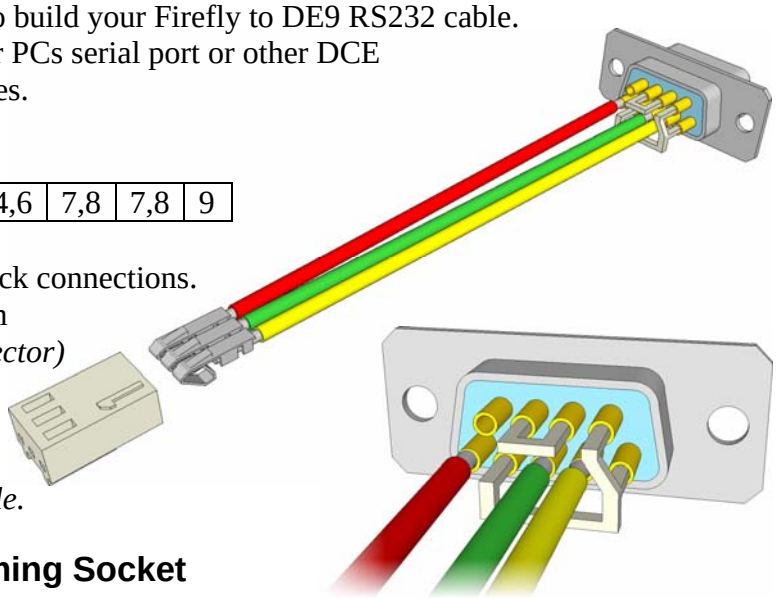
SW5-1	VR1	(0-5V pot) = up, USER port RA1 I/O = down
SW5-2	VR2	(0-5V pot) = up, USER port RA3 I/O = down
SW5-3	IR IN	(38KHz IR) receiver on RB0 = up, Pushbutton 1 is always enabled
SW5-4	Tutor	(ICD2 controls U1, 16F88), down allows 16F88 free running mode & *SW1 RESET*
SW5-5	28/40	(ICD2 setting for programming 28 or 40 pin 5v tolerant PICs)
SW5-6	8/18	(ICD2 setting for programming 8, 14, 18 or 20 pin 5v tolerant PICs)

Reset switch SW1 will function when SW5-4 is down (off). Normally when you're working with an ICD2 you control reset via MPLAB.

Firefly RS232 Cable (DTE)

The illustration on the right should help you to build your Firefly to DE9 RS232 cable. This type of cable can connect directly to your PCs serial port or other DCE *Data Computer Equipment* terminals or devices.

Firefly		1	2		3				
DE9 Female	1,4,6	2	3	1,4,6	5	1,4,6	7,8	7,8	9



The white wires in the close-up are for loopback connections.
Use shielded cable for lengths over 1' or 30cm
(Solder the shield conductor to the DE9 connector)
Note: the colors shown are for reference only.
Your cables colors may be different.

It's a good idea to use a DE9 shell on the cable.

ZIF (Zero Insertion Force) Programming Socket

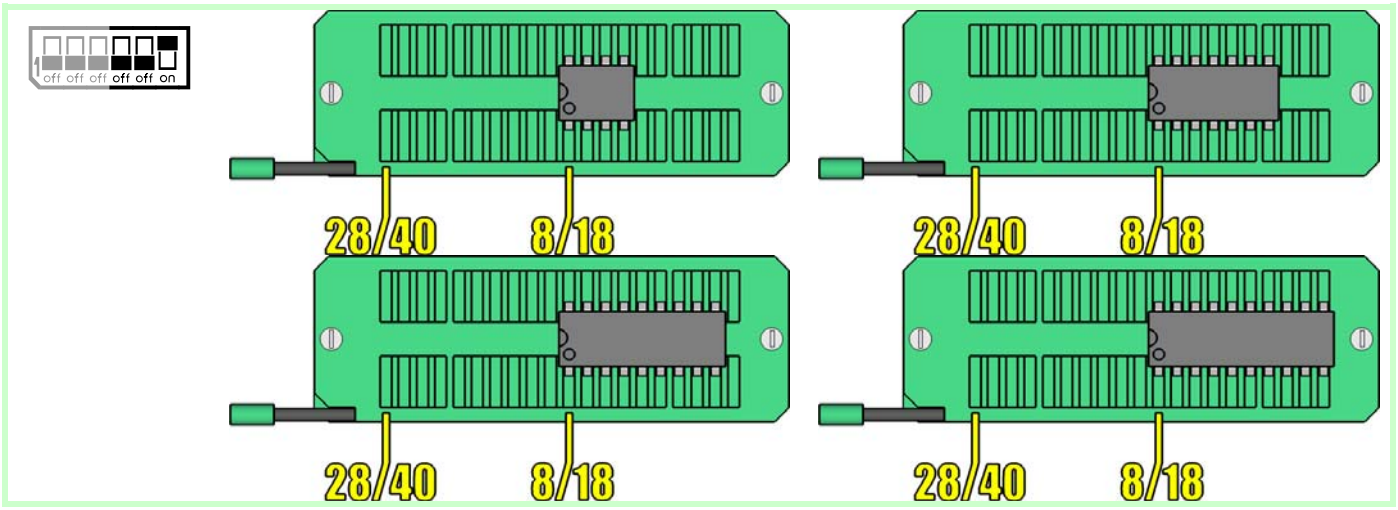
The ZIF socket option when combined with an Inchworm ICD2 is compatible with the following PICs. PICs with 20pins or less are inserted into the socket at pin#11, it's recommended to mark this pin location on the socket with a silver sharpie marker or a small dash of paint (see the cover drawing for suggestions)

ZIF Socket Compatible PICs with 0.6" & 0.3" wide DIP

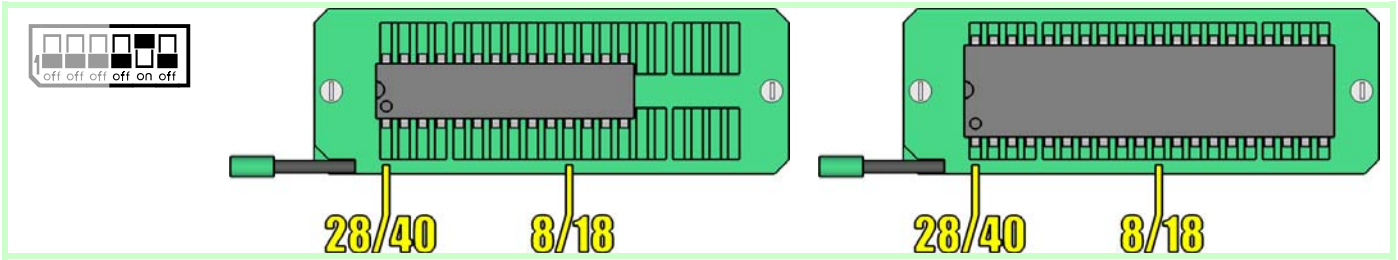
12F508, 12F509, 12F510, 12F629, 12F635, 12F675, 12F683
16Fxxxx series : 505, 506, 54, 616, 627x, 628x, 630, 631, 636, 648A, 676, 677, 684, 685, 687, 688, 689, 690, 716, 72, 73, 737, 74, 747, 76, 16F767, 16F77, 16F777, 16F785, 16F818, 819, 83, 84A, 87, 870, 871, 872, 873x, 16F874x, 876x, 877x, 88, 883, 884, 886, 887, 913, 914, 916, 917, HV616
18Cxxxx series : 242, 252, 442, 452
18Fxxxx series : 1220, 1230, 1320, 1330, 2220, 2221, 2320, 2321, 2331, 2410, 242, 2420, 2431, 2439, 2450, 2455, 248, 2480, 2510, 2515, 252, 2520, 2525, 2539, 2550, 258, 2580, 2585, 2610, 2620, 680, 2682, 2685, 4220, 4221, 4320, 4321, 4331, 4410, 442, 4420, 4431, 4439, 4450, 4455, 448, 4480, 4510, 4515, 452, 4520, 4525, 4539, 4550, 458, 4580, 4585, 4610, 4620, 4680, 4682, 4685

The Programming Switches (VPP select) and the ZIF socket

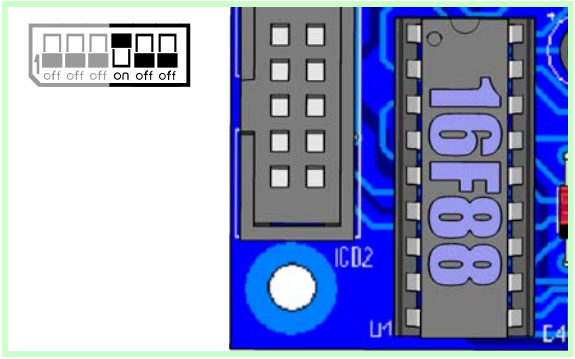
Programming 8/14/18/20 DIP PICs *(Tutor off, 28/40 off, 8/18 on)*



Programming 28/40 pin PICs *(Tutor off, 28/40 on, 8/18 off)*



Programming the Tutor 16F88 *(Tutor off, 28/40 on, 8/18 off)*




```
;*** sleep16F88.asm
    LIST p=16F88
    INCLUDE "p16f88.inc"
    __CONFIG _CONFIG1, 0x3F7C
    banksel TRISA
    movlw   b'00111111'
    movwf   TRISA
    banksel PORTA
    movlw   0x80
    movwf   PORTA  ;LED2 red
    clrwdt
    goto    $-1
    END
```

Free Running the Tutor 16F88 and Tutor / ZIF Interference

Depending on the program running on 16F88 (Tutor); it's possible to interfere with the ZIF socket.
The program "`sleep16F88.asm`" should correct the problem when using the ZIF socket.
If this fails you can hold down the reset button while using the ZIF socket or remove the 16F88.

16F88 Instruction Set

Mnemonic	Description	Operation	Status bits	
ADDLW k	Add literal and W	w + k → destination	C, DC, Z	1
ADDWF f, d	Add W and f	w + f → destination	C, DC, Z	1
ANDLW k, d	AND literal and W	w and k → destination	Z	1
ANDWF f, d	AND W and f	w and f → destination	Z	1
BCF f, b	Bit Clear f	0 → f<b>		1
BSF f, b	Bit Set f	1 → f<b>		1
BTFSC f, b	Bit Test f, Skip if Clear	skip if f<b> = 0 *(2 Cycles)*		1
BTFSS f, b	Bit Test f, Skip if Set	skip if f<b> = 1 *(2 Cycles)*		1
CALL k	Call Subroutine	PC → TOS, k → PC[10:0] PCLATH[4:3] → PC[12:11]		2
CLRF f	Clear f	0x00 → f, 1 → Z	Z	1
CLRW	Clear W	0x00 → w, 1 → Z	Z	1
CLRWDT	Clear Watchdog Timer	0x00 → WDT, 1 → TO, 1 → PD	TO, PD	1
COMF f, d	Compliment f	f - 0xFF → destination	Z	1
DECF f, d	Decrement f	f - 1 → destination	Z	1
DECFSZ f, d	Decrement f and Skip if Zero	f - 1 → destination skip if result = 0 *(2 Cycles)*		1
GOTO k	Go to address	k → PC[10:0] PCLATH[4:3] → PC[12:11]		2
INCF f, d	Increment f	F + 1 → destination	Z	1
INCFSZ f, d	Increment f and Skip if Zero	F + 1 → destination skip if result = 0 *(2 Cycles)*		1
IORLW k, d	Inclusive OR literal with W	w or k → destination	Z	1
IORWF f, d	Inclusive OR W with f	w or f → destination	Z	1
MOVF f, d	Move f	f → destination	Z	1
MOVLW k	Move literal to W	k → w		1
MOVWF f	Move W to f	w → f		1
NOP	No Operation	No Operation		1
RETFIE	Return from interrupt	TOS → PC, 1 → GIE		2
RETLW k	Return with literal in W	k → w, TOS → PC		2
RETURN	Return from Subroutine	TOS → PC		2
RLF f, d	Rotate Left f through Carry	C << f << C → destination	C	1
RRF f, d	Rotate Right f through Carry	C >> f >> C → destination	C	1
SLEEP	Enter Standby Mode	0x00 → WDT, 1 → TO, 0 → PD	TO, PD	1
SUBLW k	Subtract W from literal	k - w → destination	C, DC, Z	1
SUBWF f, d	Subtract W from f	f - w → destination	C, DC, Z	1
SWAPF f, d	Swap nibbles in f	f[3:0] → destination [7:4] f[7:4] → destination [3:0]		1
XORLW k, d	Exclusive OR literal with W	w xor k → destination	Z	1
XORWF f, d	Exclusive OR W with f	w xor f → destination	Z	1

k = 8 bit Literal or 11 bit Address, f = File Register, d = Destination (w *or* f) if omitted will default to f
skip instructions add an extra cycle if condition is true (BTFSC, BTFSS, DECFZ, INCFZ)

MPASM Pseudo Instruction Set *(mini macros for the midrange PIC)*

Mnemonic	Description	Operation	Status bits
ADDCF f, d	Add Carry to File Register	btfsc STATUS, C incf f, d	Z
ADDDCF f, d	Add Digit Carry to File Register	btfsc STATUS, DC incf f, d	Z
BC k	Branch on Carry	btfss STATUS, C goto k	
BNC k	Branch on Not Carry	btfsc STATUS, C goto k	
BDC k	Branch on Digit Carry	btfss STATUS, DC goto k	
BNDC k	Branch on Not Digit Carry	Btfsc STATUS, DC goto k	
BZ k	Branch on Zero	btfss STATUS, Z goto k	
BNZ k	Branch on Not Zero	btfsc STATUS, Z goto k	
CLRC	Clear Carry	bcf STATUS, C	C
CLRDC	Clear Digit Carry	bcf STATUS, DC	DC
CLRZ	Clear Zero	bcf STATUS, Z	Z
LCALL k	Long Call	bcf/bsf STATUS, RP0 bcf/bsf STATUS, RP1 call k	
LGOTO k	Long Goto	bcf/bsf STATUS, RP0 bcf/bsf STATUS, RP1 goto k	
MOVFW f	Move File Register to W	movf, 0	Z
NEGF f, d	Negate File Register	comf f, 1 incf f, d	Z
SETC	Set Carry	bsf STATUS, C	C
SETDC	Set Digit Carry	bsf STATUS, DC	DC
SETZ	Set Zero	bsf STATUS, Z	Z
SKPC	Skip on Carry	btfss STATUS, C	
SKPNC	Skip on Not Carry	btfsc STATUS, C	
SKPDC	Skip on Digit Carry	btfss STATUS, DC	
SKPNDC	Skip on Not Digit Carry	btfsc STATUS, DC	
SKPZ	Skip on Zero	btfss STATUS, Z	
SKPNZ	Skip on Not Zero	btfsc STATUS, Z	Z
SUBCF f, d	Subtract Carry from File Register	btfsc STATUS, C decf f, d	Z
SUBDCF f, d	Subtract Digit Carry from File Register	btfsc STATUS, DC decf f, d	Z
TSTF	Test File Register for Zero	movf f, 1	Z

Firefly 16F88 Special File Registers & RAM

The following tables are designed to supplement the 16F88 datasheet by Microchip.

	Green = bit set on POR "1" (Power On Reset)
	White = bit clear on POR "0"
	Yellow = unknown on POR (not cleared or set)
	X = not in use, usually returns a zero "0"
Italic	Italic = bit is read only
	RAM unknown state on reset (not cleared or set on POR)
	Red = Firefly Hardware does not support this feature

Common to banks 0, 1, 2 & 3 (all banks)

	Bit 7	6	5	4	3	2	1	0
FSR	Indirect Data Memory(RAM or SFR) Address Pointer							
INDF	Contents of Data Memory(RAM or SFR) Pointed to by FSR							
INTCON	GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PCL	Program Counter (LSB)							
PCLATH				Write Buffer for Upper five PCL bits				
STATUS	IRP	RP1	RP0	*NOT_TO*	*NOT_PD*	Z	DC	C
RAM 0x070 - 0x07F	STATUS xxxxxxxx							

Note: the ICD2 debugger mode uses RAM address 0x070 and 0x0F0

Specific to bank 0 (PORTB & TMR0 repeated in bank 2)

	Bit 7	6	5	4	3	2	1	0
ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO		ADON
ADRESH	A/D Result Register (MSB)							
CCP1CON			CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
CCPR1H	Capture/Compare/PWM Register 1 (MSB)							
CCPR1L	Capture/Compare/PWM Register 1 (LSB)							
PIR1		ADIF	*RCIF*	*TXIF*	*SSPIF*	CCP1IF	TMR2IF	TMR1IF
PIR2	OSFIF	CMIF		EEIF				
PORTA	Bit 4 is Zero on BOR ->							
PORTB			<- Bits 7 & 6 are Zeroed on BOR					
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
RCREG	USART Receive Data Register							
SSPBUF	Synchronous Serial Port Receive/Transmit Buffer							
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
T1CON		*T1RUN*	T1CKPS1	T1CKPS0	T1OSCEN	NOT_T1SYNC	TMR1CS	TMR1ON
T2CON		TOUTPS3	TOUTPS2	POUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
TMR0	Timer0 Module Register							
TMR1H	Holding Register for 16-bit Timer1 (MSB)							
TMR1L	Holding Register for 16-bit Timer1 (LSB)							
TMR2	Timer2 Module Register							
TXREG	USART Transmit Data Register							
RAM 0x020 - 0x06F	STATUS x00xxxxx							

Firefly 16F88 Special File Registers & RAM continued…

Specific to bank 1 (OPTION_REG & TRISB repeated in bank 3)

	Bit 7	6	5	4	3	2	1	0
ADCON1	ADFM	ADCS2	VCFG1	VCFG0				
ADRESL	A/D Result Register (LSB)							
ANSEL		ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
CVRCON	CVREN	CVROE	CVRR		CVR3	CVR2	CVR1	CVR0
OSCCON		IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
OPTION_REG	NOT_RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
OSCTUNE			TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
PCON							NOT_POR	NOT_BOR
PIE1		ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
PIE2	OSFIE	CMIE		EEIE				
PR2	Timer2 Period Register							
TRISA		*PORTA,5*	PORTA Data Direction Register					
TRISB	PORTB Data Direction Register (1 = input default), 0 = output							
TXSTA	CSRC	TX9	TXEN	SYNC		BRGH	TRMT	TX9D
SSPADD	Synchronous Serial Port Address (I2C mode)							
SSPSTAT	SMP	CKE	D_A	I2C_STOP	I2C_START	R_W	UA	BF
RAM 0x0A0 - 0x0EF	STATUS x01xxxxx							

Specific to bank 2(PORTB & TMR0 repeated in bank 2)

	Bit 7	6	5	4	3	2	1	0
EEADR	EEPROM Address Register (LSB)							
EEADRH					EEPROM Address Register (MSB)			
EEDATA	EEPROM Data Register (LSB)							
EEDATH			EEPROM Data Register (MSB)					
PORTB	<- Bits 7 & 6 are Zeroed on BOR							
TMR0	Timer0 Module Register							
WDTCON				WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN
RAM 0x110 - 0x16F	STATUS x10xxxxx							

Specific to bank 3(OPTION_REG & TRISB repeated in bank 1)

	Bit 7	6	5	4	3	2	1	0
EECON1	EEPGD			FREE	WRERR	WREN	WR	RD
EECON2	EEPROM Control Register 2 (Not a physical register)							
OPTION_REG	NOT_RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
TRISB	PORTB Data Direction Register							
RAM 0x190 - 0x1EF	STATUS x11xxxxx							

*Keep in mind pins that support A/D will default to analog mode. To configure the pins as digital I/O on Fireflys 16F88 don't forget to clear the ANSEL register bits.

```
clrf ANSEL  ;B1 will make all PORTA pins digital
```

The 16F88 configuration word

Configuration `__CONFIG` bits setup the processor what to do when initially powered up. They are often referred to as fuses and a blank or erased PIC will set the config bits *fuses* and they are "blown" at programming time. They are in memory locations that cannot be accessed or modified by 16xxx series PICs, they can only be accessed / changed by a programmer such as Inchworm.
The list below shows only settings that apply to the Firefly Tutor, not listed are any config settings that would not apply to the Fireflys 16F88. When a PIC is erased the config bits return to their default set state.

Firefly mandatory config bit settings

To ensure your Firefly will operate properly the following three config options must be enabled.

```
__CONFIG _CONFIG1, _INTRC_IO & _DEBUG_ON & _LVP_OFF
```

`_INTRC_IO` Internal RC oscillator enabled; RA6 & RA7 are available as I/O
`_DEBUG_ON` Disable RB6 & RB7 as I/O; *reserve for Inchworm use.*
`_LVP_OFF` Frees RB3 as I/O; *Inchworm is a HVP programmer / Debugger*

It's also possible to shorten the above statement to

```
__CONFIG _CONFIG1, 0x377C
```

Additional options that depend on application

To add (actually AND) more config options simply use "`&`" to AND them together.

```
__CONFIG _CONFIG1, 0x377C & _CCP1_RB3
```

It's also ok to use multiple CONFIG lines.

```
__CONFIG _CONFIG1, _WDT_OFF & _DEBUG_ON
__CONFIG _CONFIG1, _CCP1_RB0 & _PWRTE_ON
```

Default options are indicated with an asterisk.

`_CCP1_RB3` CCP1 "PWM" (RB3/CON1 pin 3)
`_CCP1_RB0` ***default** (IR_IN & SW1) handy for IR decoding "Capture"

`_WDT_ON` ***default** Enables the Watchdog Timer
`_WDT_OFF` Disable Watchdog Timer (required for debug)

Optional settings for CONFIG1

`_CP_OFF` ***default** code protect off
`_CP_ALL` code protect on *(PIC can only be erased and reprogrammed, read not possible)*
`_WRT_PROTECT_OFF` ***default** no program memory is write protected
`_WRT_PROTECT_256` first ¼ K program memory is write protected
`_WRT_PROTECT_2048` first ½ program memory is write protected
`_WRT_PROTECT_ALL` all program memory is write protected
`_CPD_ON` EEPROM data is code protected
`_CPD_OFF` ***default** EEPROM data code protect off
`_BODEN_ON` ***default** brownout detect enabled
`_BODEN_OFF` brownout detect disabled
`_PWRTE_ON` power on timer enabled *(caution do not use when ICD2 debugging)*
`_PWRTE_OFF` ***default** power on timer disabled

Although the 16F88 has CONFIG2 fuse settings it is not used by Firefly and can be safely ignored.

All MPASM assembler programs should…

- be [Tab] indented except for labels, comments
- start with a processor directive
- use the include file for that processor
- finally end with the END directive

```
; indenting is important
      list     p=16F88
      include "p16F88.inc"
      END
```

Compare and Jump if…

	X = RAM	X = LITERAL

CJA
Compare X to Y and jump if above
X > Y
	movf X, w
	subwf Y, w
	bnc ADDR
		movlw X
		subwf Y, w
		bnc ADDR

CJAE
Compare X to Y and jump if above or equal
X >= Y
	movf X, w
	subwf Y, w
	bc ADDR
		movlw X
		subwf Y, w
		bc ADDR

CJB
Compare X to Y and jump if below
X < Y
	movf Y, w
	subwf X, w
	bnc ADDR
		movlw X
		subwf Y, w
		bnc ADDR

CJBE
Compare X to Y and jump if below or equal
X <= Y
	movf Y, w
	subwf X, w
	bc ADDR
		movf Y, w
		sublw X
		bc ADDR

CJE Compare X to Y and jump if equal X = Y	movf X, w xorwf Y, w bz ADDR	movlw X xorwf Y, w bz ADDR
CJNE Compare X to Y and jump if not equal X <> Y	movf X, w xorwf Y, w bnz ADDR	movlw X xorwf Y, w bnz ADDR

I've used the MPASM pseudo instruction "b" branch is used to simplify the above statements. Pseudo instructions will simply insert the code required at assembly time. In my opinion pseudo instructions can improve readability and shorted the apparent length of a program.

It's also possible to make any of the above functions into macros. For example: the CJA macro on the right.

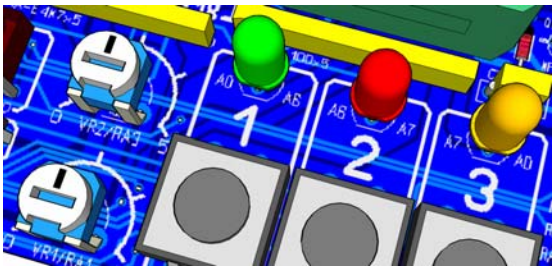
```
;*** Compare and Jump if Above
CJA macro   X, Y, ADDRESS
    movf    X, w
    subwf   Y, w
    bnc     ADDRESS
    endm
```

To use the CJA macro after it's declared simply use a statement similar to:

```
CJA cats, dogs, destination      ; if cats > dogs then goto destination
```

Firefly 16F88 Project:

"Blinky VR1"

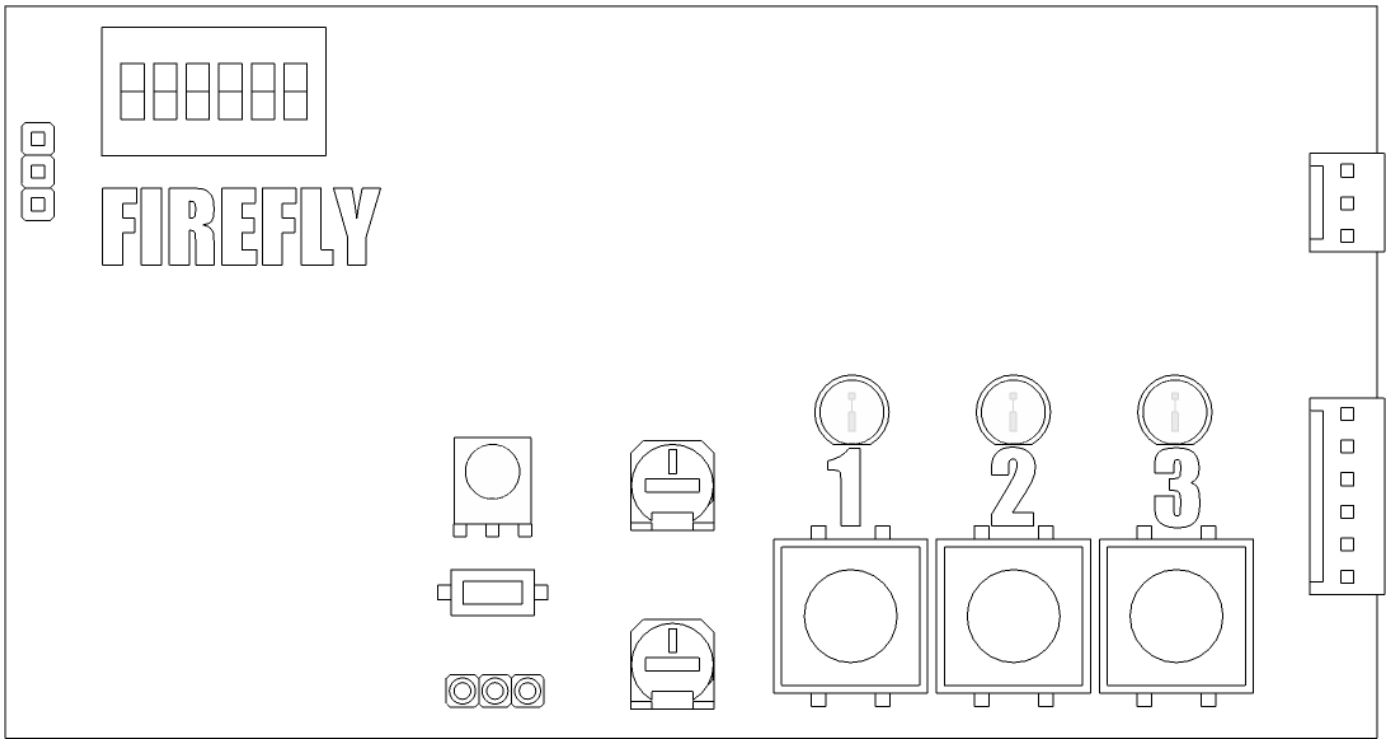


This program will flash the three bicolor (Red/Green) LEDs in a back and forth fashion on your Firefly. You can adjust the speed with the potentiometer VR1.

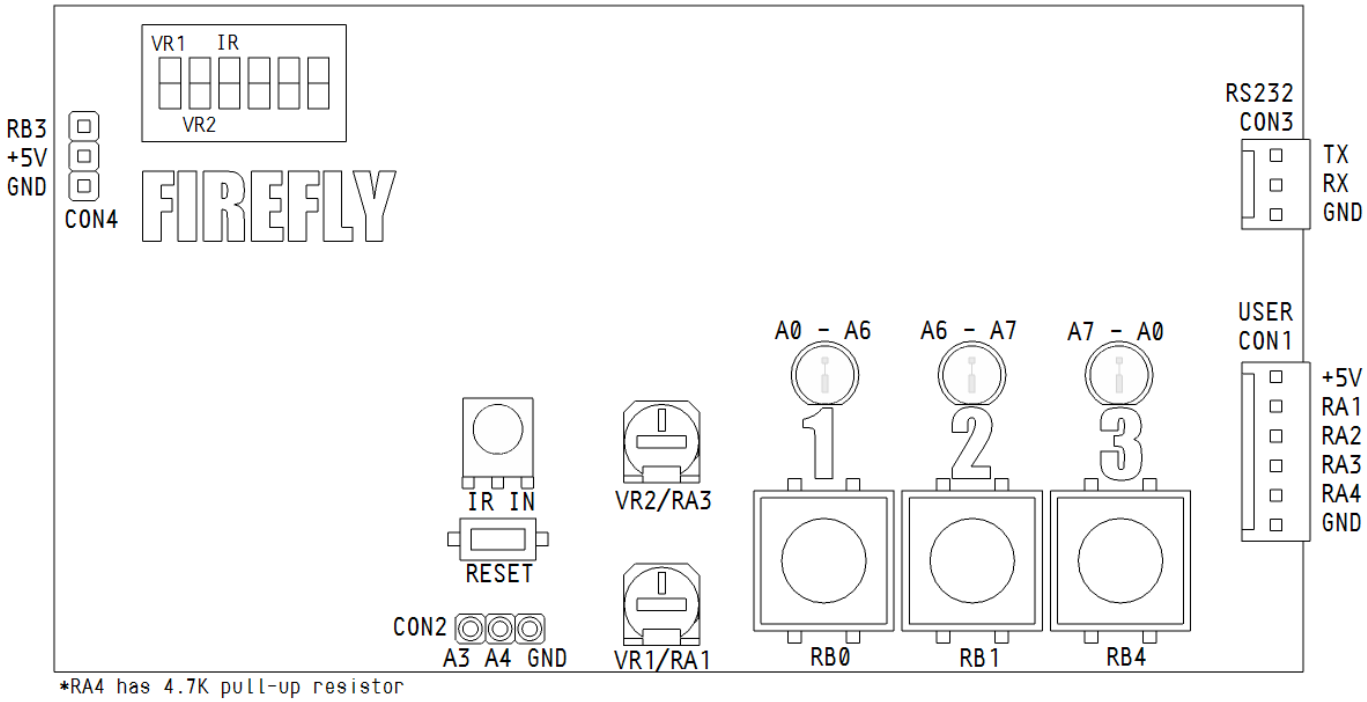
```
;*** Blinky VR1 requires the use of VR1 (DIP switch VR1 = ON, Tutor = ON, all others OFF/down)
;*** LEDs 1 thru 3 will flash back and forth with the rate controlled by VR1
        LIST    p=16F88                 ;
        INCLUDE <p16f88.inc>            ;
        ERRORLEVEL 0, -302              ; suppress bank select warning
        __CONFIG _CONFIG1, 0x2F30

_let    macro   reg, lit                ; MACRO _let <register>, <literal>
        movlw   lit                     ; W = literal
        banksel reg                     ; make sure it's in the right bank
        movwf   reg                     ; register = W
        endm

Count   equ     0x20                    ; delay loop counter (0x20 is RAM start location)
        org     0x000                   ; this is the reset vector
        nop                             ; nop if you plan on using the debugger
Init    _let    ANSEL, b'00000010'      ; select RA1/AN1 as analog input
        _let    ADCON0, b'11001101'     ; AD enabled, AN1 selected (VR1) & begin conversion

Red1    _let    PORTA, b'01000000'      ; LED1 RED
        _let    TRISA, b'10111110'      ; R x x
        call    Delay
Red2    _let    PORTA, b'10000000'      ; LED2 RED
        _let    TRISA, b'00111111'      ; x R x
        call    Delay
Red3    _let    PORTA, b'00000001'      ; LED3 RED
        _let    TRISA, b'01111110'      ; x x R
        call    Delay
Green3  _let    PORTA, b'10000000'      ; LED3 GREEN
        _let    TRISA, b'01111110'      ; x x G
        call    Delay
Green2  _let    PORTA, b'01000000'      ; LED2 GREEN
        _let    TRISA, b'00111111'      ; x G x
        call    Delay
Green1  _let    PORTA, b'00000001'      ; LED1 GREEN
        _let    TRISA, b'10111110'      ; G x x
        call    Delay
        goto    Red1                    ; repeat forever

Delay   ;*** Rotating VR1 varies loop delay, fast flashing LEDs will appear Orange
        bcf     STATUS, RP0             ;B0 contains ADRESH, ADCON0 & Count
        movf    ADRESH, W               ;B0 W = VR1 value 0x00-0xFF (also tests for zero)
        movwf   Count                   ;B0 let Count = W
        bsf     ADCON0,GO               ;B0 start next conversion (Set A/D go/done bit)
        bz      Exit                    ; if Count = 0 then return early (no need to delay)
Loop    ;*** The main delay loop begins here
        nop                             ; stretches out the delay routine
        decfsz  Count                   ;B0 Count = Count - 1, if result is zero then skip
        goto    Loop                    ; not yet zero so loop
Exit    return                          ; return
        END
```

Firefly Templates

The following templates are designed to aid in documenting your own firefly projects. It is simplified to show only available built in devices, external I/O ports & switch settings. You can use these templates for your own Firefly projects and documentation.



Both templates can be downloaded in PNG format and used with your Fireflys' project documentation from http://www.blueroomelectronics.com

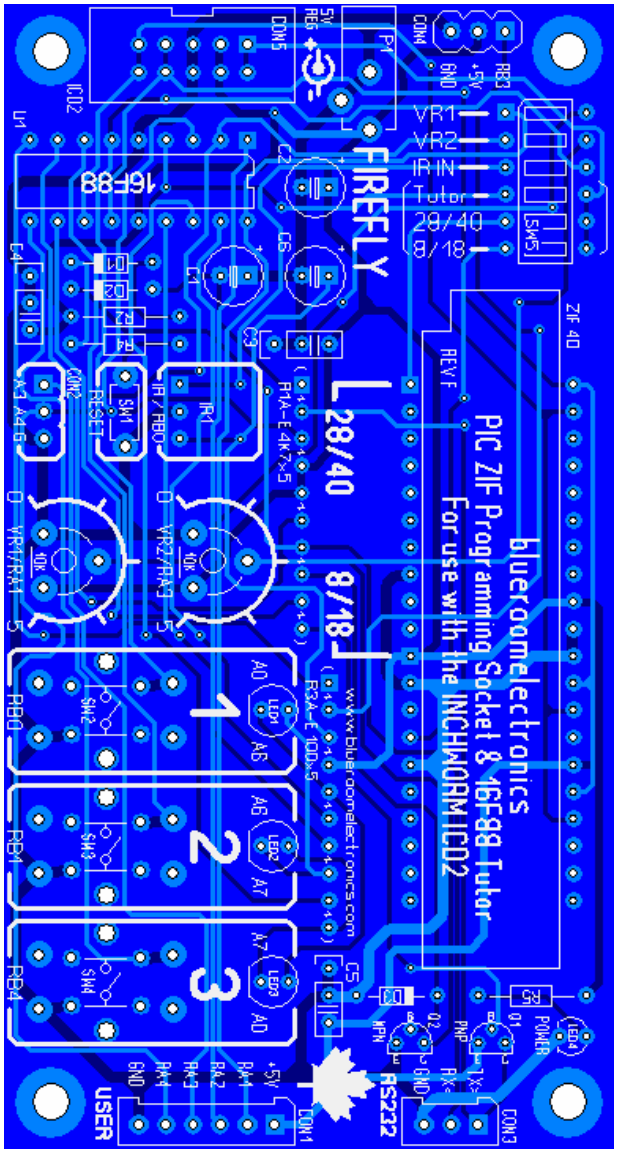


Retail Sales

CREATRON INC.
255 College St. Toronto
Ontario, Canada
Tel (416) 977-9258
Fax (416) 977-4700
creatronpart@hotmail.com
http://www.creatroninc.com

Online Sales

dipmicro electronics
1251 Walkers Line, Burlington
Ontario, Canada L7M4N8
Fax (866) 603-7109
http://www.dipmicro.com



Dealer Sales & Technical Inquiries

blueroomelectronics
4544 Dufferin St. Toronto
Ontario, Canada
Tel (416) 897-1962
sales@blueroomelectronics.com
http://www.blueroomelectronics.com

Info and all other inquiries info@blueroomelectronics.com