2009

Introduction to Microcontrollers

Intersession 2009

[INTRODUCTION TO MICROCONTROLLERS]

[Information Packet – babaker2@illinois.edu – sturnfie@illinois.edu – thoulah2@illinois.edu]

Contents

Reference Material 4

PIC 16F877A Pinout 4

Bank Bits 4

PIC 16F877A F Register Map 5

PIC 16F877A Analog Port Configuration 6

PIC Instruction Set 7

MAX232 Pinout 8

ASCII Table 9

Resistor Color Code 10

Other Microcontrollers 12

Atmel AVR 12

Make Controller Kit 12

Arduino 12

PIC-based micros 12

BASIC Stamp 12

PICAXE 12

Propeller 13

ARM 13

Internet Resources 14

Parts Suppliers 14

Tutorials & Reference 15

Community, Inspiration, & Cool Stuff 16

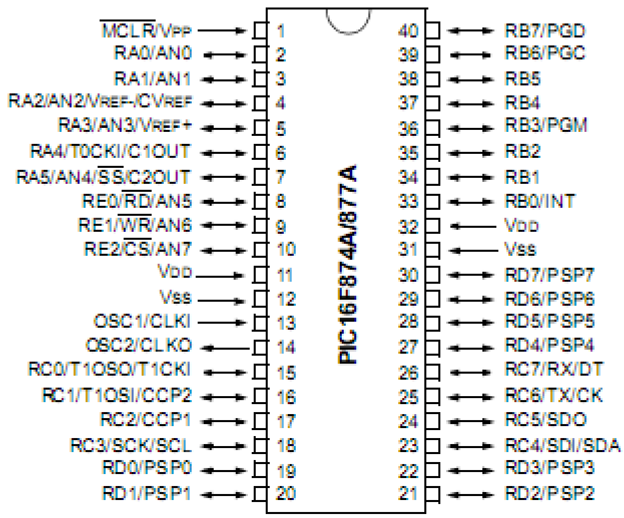
Contact Information 18

Plug for Electronics @ IMSA 18

Reference Material

Pinout, Bank bits, F-registers, and Analog Port Config, and Instructions are all taken from PIC16F877A datasheet from Microchip.
MAX232 pinout is from MAX232 datasheet.
ASCII table is from ASCIItable.com

PIC 16F877A Pinout



Bank Bits

RP1:RP0	Bank
00	0
01	1
10	2
11	3

PIC 16F877A F Register Map

Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3	File Address
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD(1)	08h	TRISD(1)	88h		108h		188h
PORTE(1)	09h	TRISE(1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved(2)	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved(2)	18Fh
T1CON	10h		90h	General Purpose Register 16 Bytes	110h–11Fh	General Purpose Register 16 Bytes	190h–19Fh
TMR2	11h	SSPCON2	91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch	CMCON	9Ch				
CCP2CON	1Dh	CVRCON	9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
General Purpose Register 96 Bytes	20h–7Fh	General Purpose Register 80 Bytes	A0h–EFh	General Purpose Register 80 Bytes	120h–16Fh	General Purpose Register 80 Bytes	1A0h–1EFh
		accesses 70h-7Fh	F0h–FFh	accesses 70h-7Fh	170h–17Fh	accesses 70h - 7Fh	1F0h–1FFh

Unimplemented data memory locations (shaded), read as '0'.

* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876A.

2: These registers are reserved; maintain these registers clear.

PIC 16F877A Analog Port Configuration

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O
C/R = # of analog input channels/# of A/D voltage references

PIC Instruction Set

TABLE 15-2: PIC16F87XA INSTRUCTION SET

Mnemonic,	Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
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	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

MAX232 Pinout

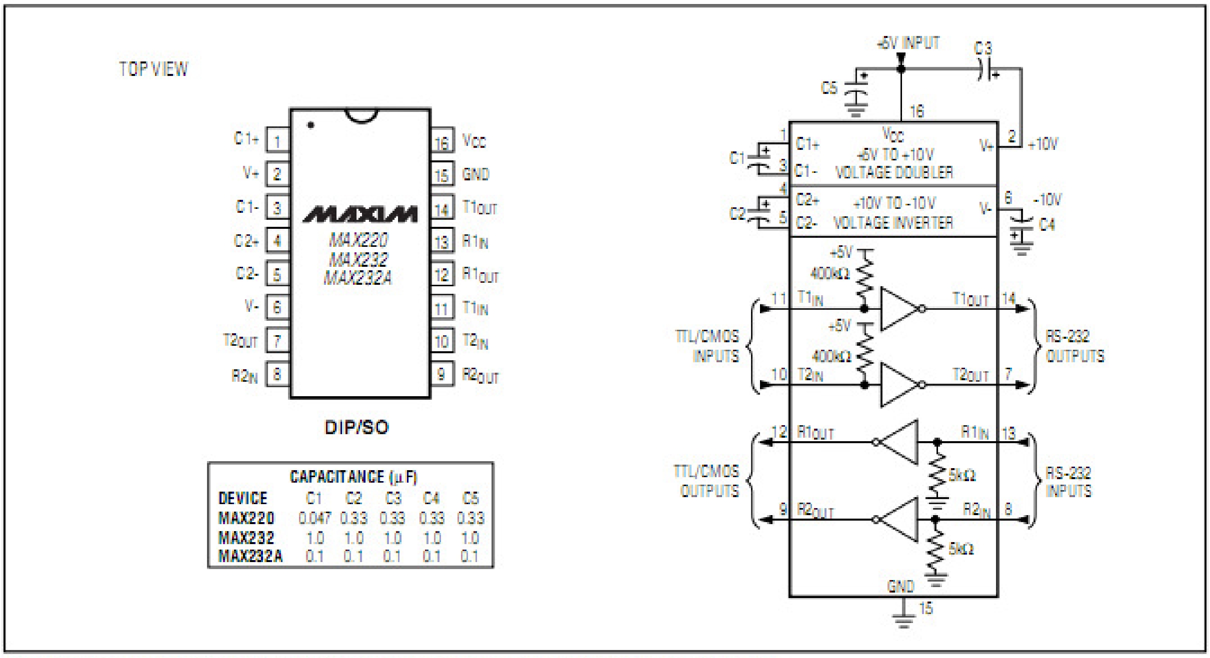


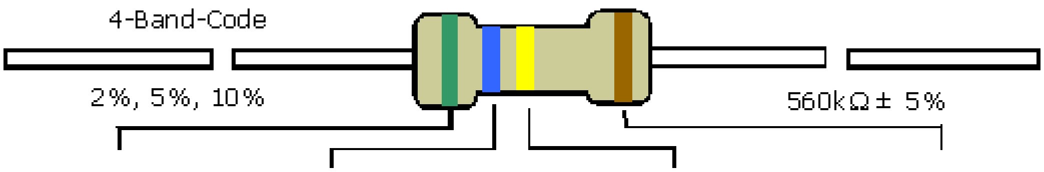
Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

ASCII Table

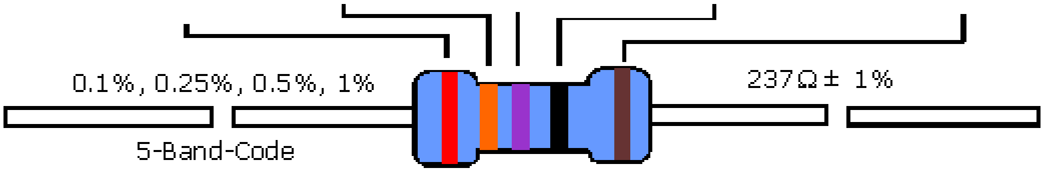
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Resistor Color Code



COLOR	1st BAND	2nd BAND	3rd BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	±0.5% (D)
Blue	6	6	6	1MΩ	±0.25% (C)
Violet	7	7	7	10MΩ	±0.10% (B)
Grey	8	8	8		±0.05%
White	9	9	9		
Gold				0.1	± 5% (J)
Silver				0.01	± 10% (K)



Other Microcontrollers

Atmel AVR

AVR Microcontrollers are very comparable to PICs; they are 8-bit RISC single chip microcontrollers, supporting numerous features like USART, I2C, PWM, etc, run at about the same speeds, etc. Programming AVRs in assembly may be easier than for PICs, because you can move data around directly from register to register without having to always pass through the working register, and you don't have to deal with bank bits. AVR also has a free cross-platform C compiler (with full optimization), and it makes porting code between different models easier. However, AVRs aren't as common as PICs, so tutorials and info online is harder to come by. If you see a part number prefixed with ATiny or ATmega, followed by a number, this is an AVR chip.

Make Controller Kit

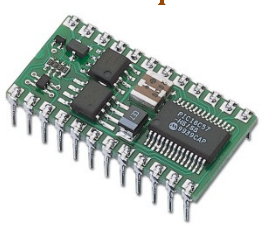
The MAKE Controller is a development board produced by Make Magazine. It is powered by an AVR chip.

Arduino

Arduino is an open-source development board powered by an ATmega. It is programmed in the Processing language with the Wiring extension. Processing is comparable to C and Java, but includes a lot of visual elements and is targeted towards artists. As a result, many electronic and interactive arts projects use an Arduino.

PIC-based micros

BASIC Stamp



The "BASIC" part of the name comes from the programming language used; a special version of BASIC is tokenized and downloaded to the stamp; instead of being compiled to machine code, the BASIC commands are interpreted. The "Stamp" part of the name comes from the physical structure of it; the stamp is a small circuit board that is the same shape as a through-hole IC. The Stamp includes a voltage regulator, oscillator, EEPROM, a PIC to interpret the BASIC code, and other supporting circuitry. The Basic Stamp comes in two versions; BS1 and BS2, and the BS2 has many subversions. The BASIC Stamp is produced and sold by Parallax (http://parallax.com).

PICAXE

The PICAXE is a lot like the BASIC Stamp without the Stamp part; it's essentially a PIC with a bootloader than downloads and interprets BASIC instructions.

Propeller

The Propeller is another microcontroller from Parallax. It was designed, developed, and is manufactured completely by Parallax – in fact, the Parllax president Chip Gracey developed the micro, assembly language, and Spin language by himself. In contrast to the 8-bit PIC based BASIC Stamp, the Propeller is a 32-bit multi-core microcontroller; it features 8 separate CPUs, referred to as “cogs”. In contrast to PICs, not many features are supported in hardware (like UART and I2C); instead, such features are implemented in software using downloadable code libraries. This is made possible in part because SPIN is a high level object oriented language, and because of the Propeller’s raw power. One of the few hardware features the Propeller includes is precise timing structures to aid in display PAL, NTSC, and VGA video (or other timing-based tasks).

ARM

I haven’t seen ARM controllers used in many hobby applications. They are typically higher powered 32-bit micros and are used in many commercial embedded applications, such as cell phones. For example, the T-Mobile G1, the first Android phone, uses an ARM processor.

Internet Resources

Parts Suppliers

Sparkfun
http://www.sparkfun.com
Sparkfun is a store for the electronics enthusiast. They have a number of great components, including a wide selection of microcontrollers. A lot of the components we obtained for this Intersession were obtained from Sparkfun. They also have posted a number of tutorials for various microcontrollers.

Newark
http://www.newark.com
Newark has a gigantic catalog of many variations of nearly every basic component you can think of; different resistors, capacitors, and lots of chips can be found here. This website is geared more towards commercial customers, so if you're interested in learning to solder suface mount parts, you can get tons for cheap here.

Digikey
http://www.digikey.com
Digikey is a great site with tons of parts for good prices, as long as you know exactly what you're looking for. Some people prefer to look for parts with Newark's catalog, and then try looking up the part numbers in Digikey to see if they are cheaper.

Octopart
http://www.octopart.com
Octopart isn't a parts supplier per se; rather, it is a search engine for electronics parts. For each part it finds related to your search, it lists the prices from various suppliers and also provides a link to the datasheet if it can find one. I don't have much experience with it, but I must say I was very impressed by the test search I just tried!

Tutorials & Reference

Sparkfun
http://www.sparkfun.com
Sparkfun was already listed under Parts Suppliers, but they also have a bunch of tutorials on their website.

AllDataSheet.com
http://alldatasheet.com/
A great site for finding datasheets for almost any part under the sun. This is my first stop when I get a bag of unknown chips.

Instructables
http://www.instructables.com
This is a website specifically for people to post step-by-step instructions, with pictures, about how they've built various projects. Oftentimes, you'll discover yourself at Instructables after clicking a link at Hackaday or the Make Blog.

Community, Inspiration, & Cool Stuff

Hackaday
http://www.hackaday.com
Brian's person favorite hacking blog. Hackaday predominately features electronics hardware hacking, but a lot of great software and security pieces get posted, too. The name is vestigial; Hackaday has expanded recently and posts well over one hack per day.

Makezine
http://www.makezine.com
This website is for an O'Reilly magazine called *Make*, which is for the DIY crowd. The website has a blog filled with links to projects posted online. While electronics is often a part of the projects, this is aimed at a wider crowd and will feature posts based on everything from engines to origami to tanning leather to whatever else you can imagine! There's a very high volume of posts; add this to your RSS feeder and you'll never has a reason to be bored again.

Bre Pettis's blog
http://brepettis.com/blog/
Bre Pettis is probably one of the most well-known names in the hardware hacking scene. Previously, he hosted the Weekend Project video series for Make. He currently is doing a video series for Etsy (http://www.etsy.com/), a website that serves as a storefront for beautiful handcrafted items, and the pilot episode of his television show "History Hacker" has aired on the History channel. He keeps a blog with short interviews with some of his friends about the things they make, often at the NYCResistor, a hackerspace in New York of which Bre is a founding member. [If you're interested in a hackerspace in the Chicago area, I just discovered that one called Pumping Station: One (http://pumpingstationone.org) has been started. Hackerspaces are basically places hackers go to build things, and often have CNC equipment like laser cutters and Addition-based rapid prototyping machines like the RepRap available]

Hacked Gadgets
http://www.hackedgadgets.com
This site is a lot like hackaday.com. It is typically a little less technical and contains less software and security material; its focus is tighter on, well, hacked gadgets.

Others
http://hacknmod.com
http://electronics-lab.com
http://projects-lab.com
http://steampunkworkshop.com/ - Steampunk oriented, often has projects
http://brassgoggles.co.uk/brassgoggles/ - Steampunk orients, often has projects

Contact Information

Ms. Diane Hinterlong
dlh@imsa.edu
Office: B 115
Phone: x5969

Brian Baker
babaker2@illinois.edu

Lucas Sturnfield
sturnfie@illinois.edu

Tom Houlahan
thoulah2@illinois.edu

Plug for Electronics @ IMSA

Besides giving you a lot more time to practice techniques in the lab, you'll get practice with KVL & KCL and resistor networks. You'll learn more about AC to DC conversion and exactly how capacitors works. You'll learn to use the most important tool in electronics – the oscilloscope – as well a function generator. You'll learn a lot more about transistors and a bunch of other important digital circuits, like flip flops, 555 timers, op amps, and shift registers!