PURDUE UNIVERSITY DEPARTMENT OF PHYSICS

Physics 53600
Electronics Techniques for Research

Now in PowerPoint!

Spring 2020 Semester

Prof. Matthew Jones

The usual ANNOUNCEMENT

- Obvious changes to the course:
  - No in-person lectures: you'll have to read the lecture notes yourself
  - No more labs: don't worry about it – your grade will be based on work done so far
  - Remaining assignments will try to cover topics that would have been explored in the lab
  - Second mid-term: simplest to cancel it
  - Final exam: I think it will be a 24 hour exam with written responses that can be easily sent by e-mail.
- Changes to grading scheme:
  - Old scheme: Assignments (30%) exams (40%) lab (30%)
  - New scheme: Assignments (50%) exams (25%) lab (25%)

The usual ANNOUNCEMENT

- Because there won't be any in-person lectures, you will have to read the lecture notes yourself.
- To demonstrate that you have read them, you will be required to answer *one or two simple questions* before the next lecture is posted.
- The question will probably be at the beginning and you just have to e-mail me the answer

mjones@physics.purdue.edu

- To make this easy, please make your subject look like this:

"PHYS53600 Lecture xx questions Your Name"

- These will be the remaining part of your assignment grade.

More ANNOUNCEMENTS

- Feel free to send me questions about the lecture material if there is anything you don't understand. I'm happy to give more explanation (and I'm soooo bored.)
- Send me e-mail if you think it would be useful to arrange a time as a class to have a time where you can ask questions by video.
  - So far a couple of people have said it would be
  - Maybe something like Wednesday, April 30th at 10:30 am EDT?

LECTURE 26 QUESTIONS

1. *Is 10:30 am on Thursday, April 30th (the normally scheduled class time) a good time for a Webex meeting to ask any questions you might have?*

2. *Describe an application where a processor would be well suited, as opposed to programmable logic.*
3. *Describe an application where programmable logic would be better suited.*

Microprocessors

- So far we have discussed
  - Analog electronics
  - Digital electronics
- These are sometimes essential elements of data acquisition or signal processing systems
- But they are not always the best choice…
- The design process usually tries to optimize many different types of resources
  - Time (design time, implementation time)
  - Cost (design cost, cost to build, cost to maintain)
  - Performance (does the design satisfy the requirements?)
  - Available expertise

Microprocessors

- Microprocessors can be a very effective element of a data acquisition/instrumentation system
- Advantages:
  - Reprogrammable
  - High level programming languages
  - Low cost
  - System-on-Chip integrates many digital logic elements into one device
- Disadvantages:
  - Sequential program execution
  - Limited bandwidth
  - Sometimes too slow for some applications

Microprocessors

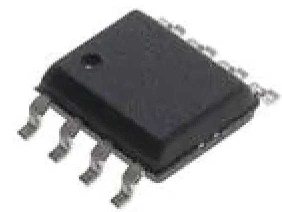
- An entire course could be dedicated to microprocessor architectures
- Historically, using a microprocessor required designing an entire computer, complete with memory and I/O interfaces
- Over the past 15 years most of these features have been integrated into processors to provide complete computer systems in one package.

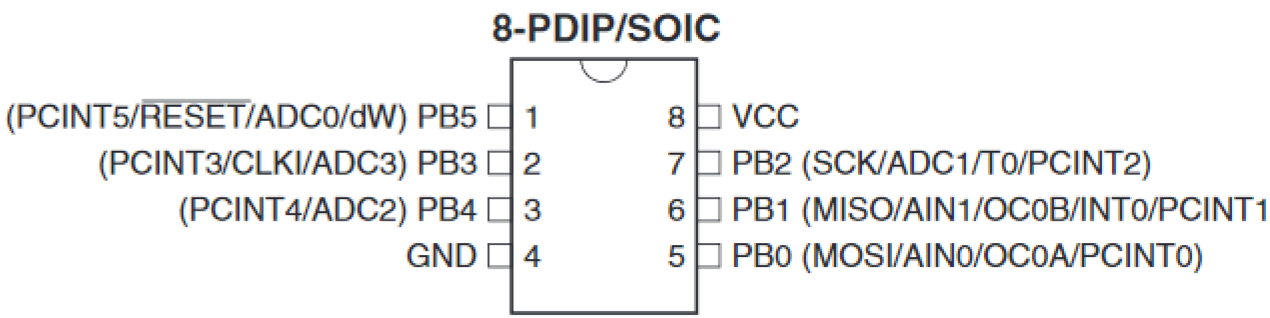
System-on-Chip Microprocessors

- Include more functionality than just the processor:
  - Electrically erasable read-only-memory (stores programs)
  - Static random-access-memory (stores data)
  - General purpose IO pins
  - Analog-to-digital converters
  - Serial communications devices
    - UART, I2C, SPI interfaces

Example (cost is < $1)

- You don’t need a lot of pins to have a fully functional SOC:
- Microchip ATTINY13A-SSU:



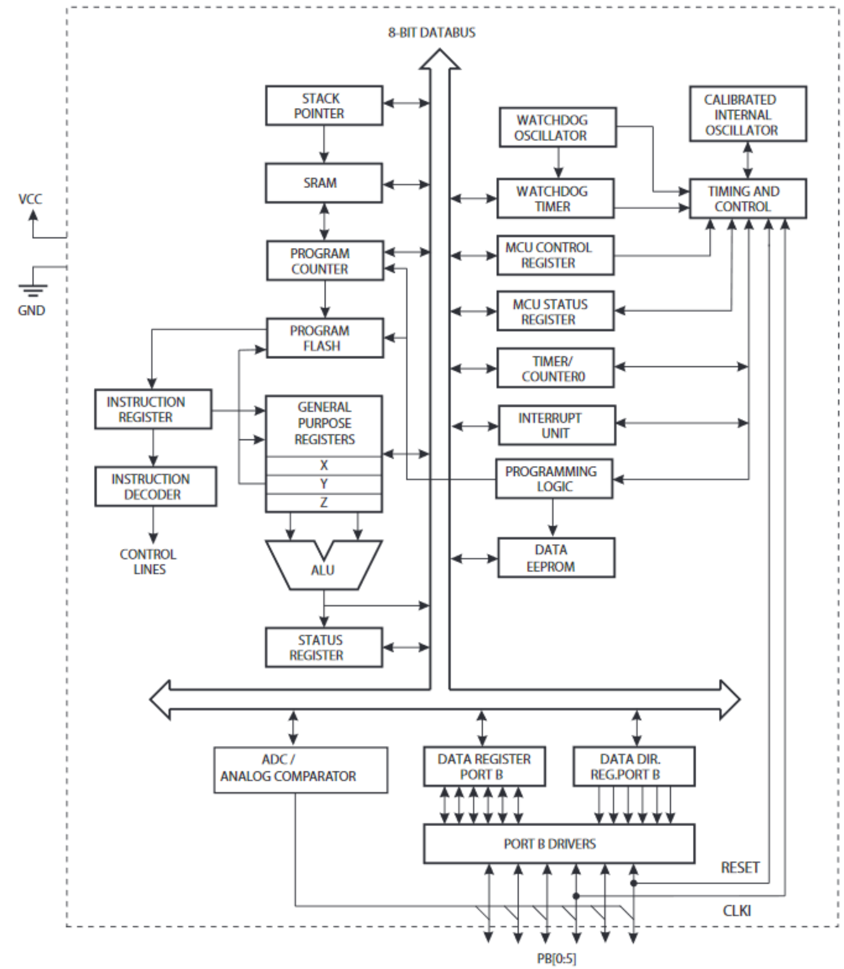


- Six I/O pins can be configured to perform various functions

Example: ATTINY13A-SSU

- There is a lot going on inside:
- Three pins implement an SPI interface that can be used to write a program to the non-volatile memory
- When powered on or reset, the processor executes the instructions stored in memory.

Figure 2-1. Block Diagram

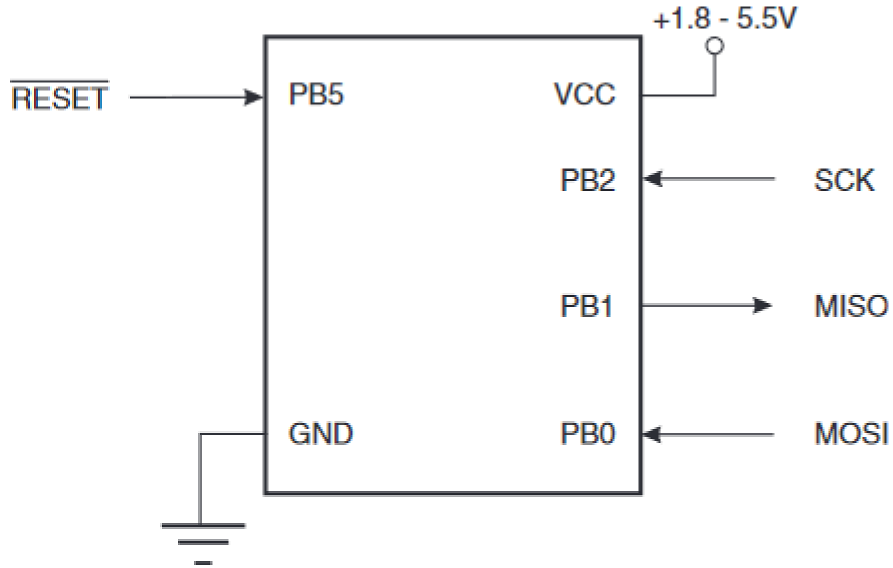


Example: ATTINY13A-SSU

17.6 Serial Programming

Both the Flash and EEPROM memory arrays can be programmed using the serial SPI
$\overline{\text{RESET}}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and M
put). See Figure 17-1.

Figure 17-1. Serial Programming and Verify



- Typically, a high-level programming language (an assembler or compiler) would be used to generate the executable program memory image.
- This would then be downloaded into the non-volatile flash memory.

Example: ATTINY13A-SSU

- The first 64 bytes of the address space control the operation of the chip:

10.4.2 PORTB – Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x18	–	–	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.4.3 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x17	–	–	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

10.4.4 PINB – Port B Input Pins Address

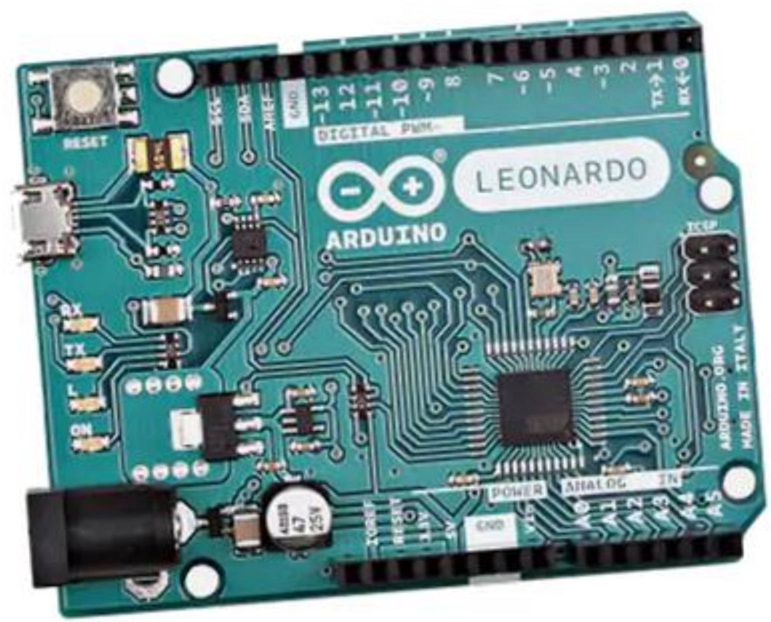
Bit	7	6	5	4	3	2	1	0	
0x16	–	–	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	N/A	N/A	N/A	N/A	N/A	

20. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x3F	SREG	I	T	H	S	V	N	Z	C	page 9
0x3E	Reserved	–	–	–	–	–	–	–	–	
0x3D	SPL	SP[7:0]								page 11
0x3C	Reserved	–	–	–	–	–	–	–	–	
0x3B	GIMSK	–	INT0	PCIE	–	–	–	–	–	page 47
0x3A	GIFR	–	INTF0	PCIF	–	–	–	–	–	page 48
0x39	TIMSK0	–	–	–	–	OCIE0B	OCIE0A	TOIE0	–	page 75
0x38	TIFR0	–	–	–	–	OCF0B	OCF0A	TOV0	–	page 76
0x37	SPMCSR	–	–	–	CTPB	RFLB	PGWRT	PGERS	SELFPR-	page 98
0x36	OCR0A	Timer/Counter – Output Compare Register A								page 75
0x35	MCUCR	–	PUD	SE	SM1	SM0	–	ISC01	ISC00	pages 33, 47, 57
0x34	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	page 42
0x33	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	page 73
0x32	TCNT0	Timer/Counter (8-bit)								page 74
0x31	OSCCAL	Oscillator Calibration Register								page 27
0x30	BODCR	–	–	–	–	–	–	BODS	BODSE	page 33
0x2F	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	page 70
0x2E	DWDR	DWDR[7:0]								page 97
0x2D	Reserved	–								
0x2C	Reserved	–								
0x2B	Reserved	–								
0x2A	Reserved	–								
0x29	OCR0B	Timer/Counter – Output Compare Register B								page 75
0x28	GTCCR	TSM	–	–	–	–	–	–	PSR10	page 78
0x27	Reserved	–								
0x26	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	page 28
0x25	PRR	–	–	–	–	–	–	PRTIM0	PRADC	page 34
0x24	Reserved	–								
0x23	Reserved	–								
0x22	Reserved	–								
0x21	WDTCR	WDTIF	WDTIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	page 42
0x20	Reserved	–								
0x1F	Reserved	–								
0x1E	EEARL	–	–	EEPROM Address Register						page 20
0x1D	EEDR	EEPROM Data Register								page 20
0x1C	EECR	–	–	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	page 21
0x1B	Reserved	–								
0x1A	Reserved	–								
0x19	Reserved	–								
0x18	PORTB	–	–	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	page 57
0x17	DDRB	–	–	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	page 57
0x16	PINB	–	–	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	page 58
0x15	PCMSK	–	–	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	page 48
0x14	DIDR0	–	–	ADC0D	ADC2D	ADC3D	ADC1D	AIN1D	AIN0D	pages 81, 95
0x13	Reserved	–								
0x12	Reserved	–								
0x11	Reserved	–								
0x10	Reserved	–								
0x0F	Reserved	–								
0x0E	Reserved	–								
0x0D	Reserved	–								
0x0C	Reserved	–								
0x0B	Reserved	–								
0x0A	Reserved	–								
0x09	Reserved	–								
0x08	ACSR	ACD	ACBG	ACO	ACI	ACIE	–	ACIS1	ACIS0	page 80
0x07	ADMUX	–	REFS0	ADLAR	–	–	–	MUX1	MUX0	page 92
0x06	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	page 93
0x05	ADCH	ADC Data Register High Byte								page 94
0x04	ADCL	ADC Data Register Low Byte								page 94
0x03	ADCSRB	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	pages 80, 95
0x02	Reserved	–								
0x01	Reserved	–								
0x00	Reserved	–								

Example ($4)

- The ATmega32U4 is used for the popular Arduino platform:
- Features:
  - 32 kB flash RAM
  - 2.5 kB static RAM
  - USB interface
  - Serial port (UART)
  - 12 channel 10-bit ADC
  - Lots more…

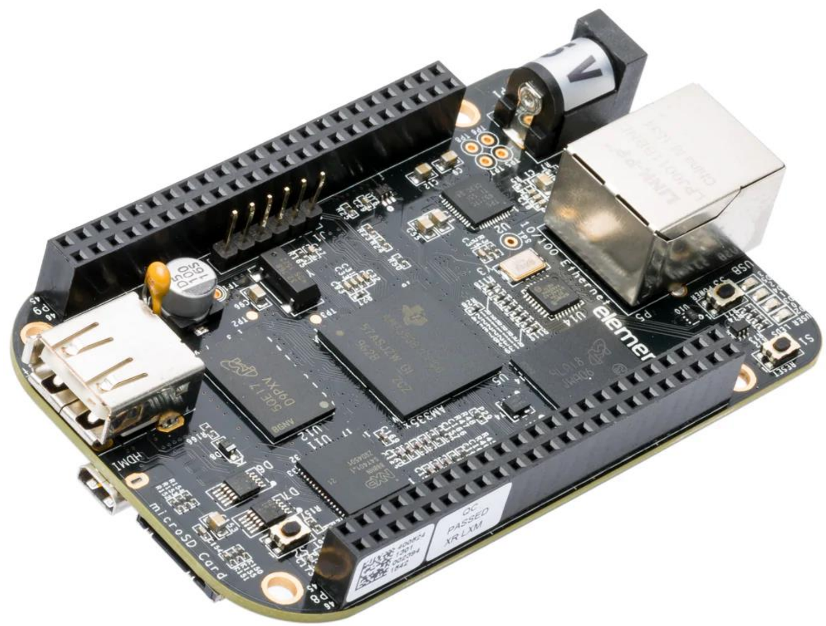


Example: ATmega32U4

- As with the previous example, the peripherals are controlled using dedicated addresses
- In this case there are 128 bytes reserved for peripheral control interfaces

Example: Beaglebone ($50)

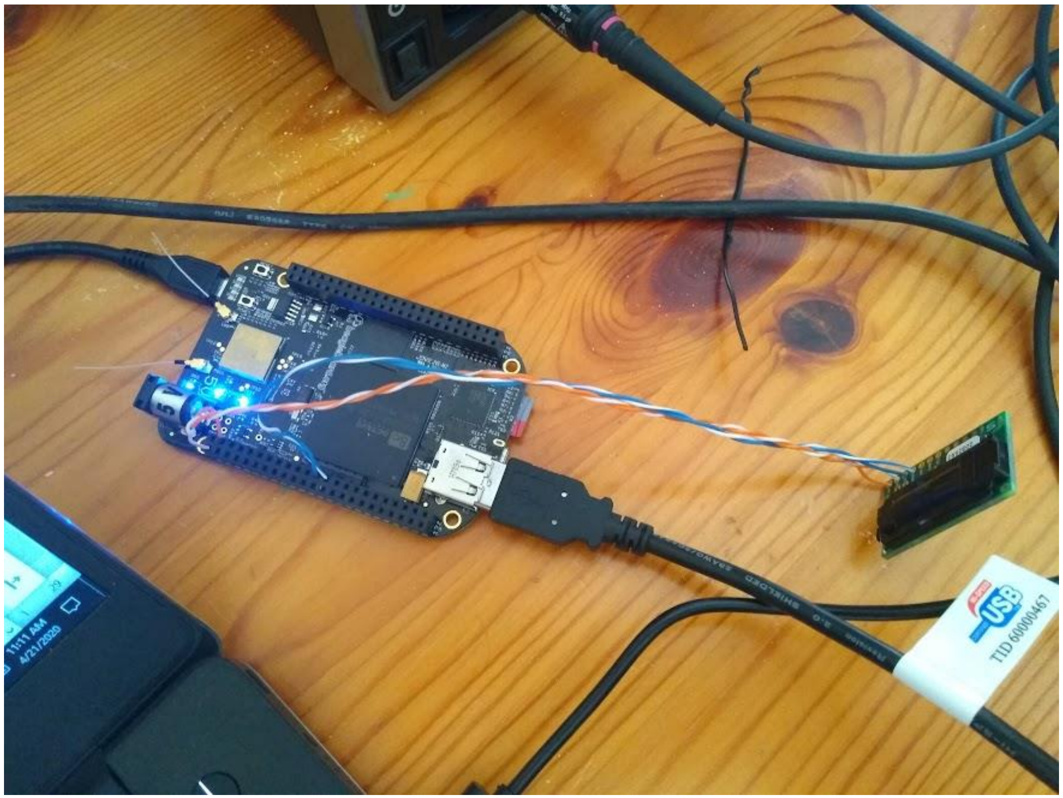
- This is a single-board computer based on the CORTEX A8 processor:
- Peripherals:
  - Ethernet
  - HDMI video
  - USB
  - GPIO pins
  - I2C, SPI, UART
  - ADC’s



Example: Beaglebone ($50)

- The CORTEX A8 processor is powerful enough to run various operating systems (Linux/Adroid)
- Typically written to a microSD memory card
- Provides everything you would expect from a Linux environment
- The Linux kernel attempts to control access to low-level hardware
- This is an open-source project… it’s free, but you get what you pay for

Example: Beaglebone



Temperature/humidity/ CO_2 sensor read out using I2C

Remote connection to processor over WiFi

Running Debian Linux distribution

RS485 network interfaced using USB

You can build quite complex data acquisition systems using inexpensive open-source hardware.

“Soft” Processors

- Microprocessors can be implemented in programmable logic and interfaced directly with FPGA resources.
- Xilinx provides free access to its MicroBlaze core designs
- Intel (which bought Altera) provides it’s version (Nios-II)

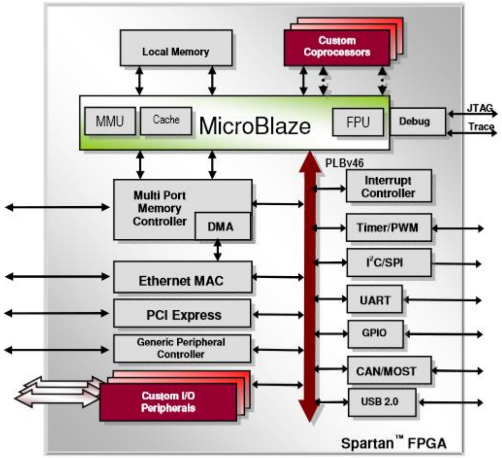
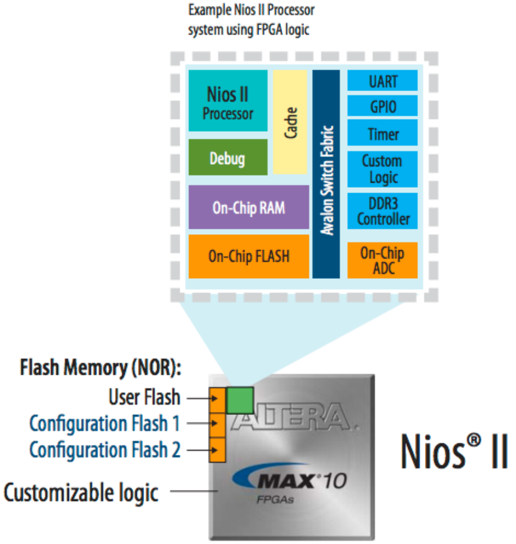


Figure 1 - MicroBlaze Processor System Architecture



Comparison of SOC and PC's

- At some point, you might really want to use a PC instead
  - Much more memory
  - Higher speeds
  - Multi-core processors
  - Higher network/memory bandwidth
- Usually there is no direct access to hardware
- Hardware interfaces must conform to the PC architecture
  - PCIe interfaces and associated software drivers
  - USB
  - Serial ports (eg. RS232)

Summary

- Today, microprocessors and system-on-chip solutions are very powerful, inexpensive, and relatively easy to use
- Some are very simple (small pin count)
- Some are sophisticated but can be bought as modules and inserted into a design
  - No need to design a sophisticated PCB