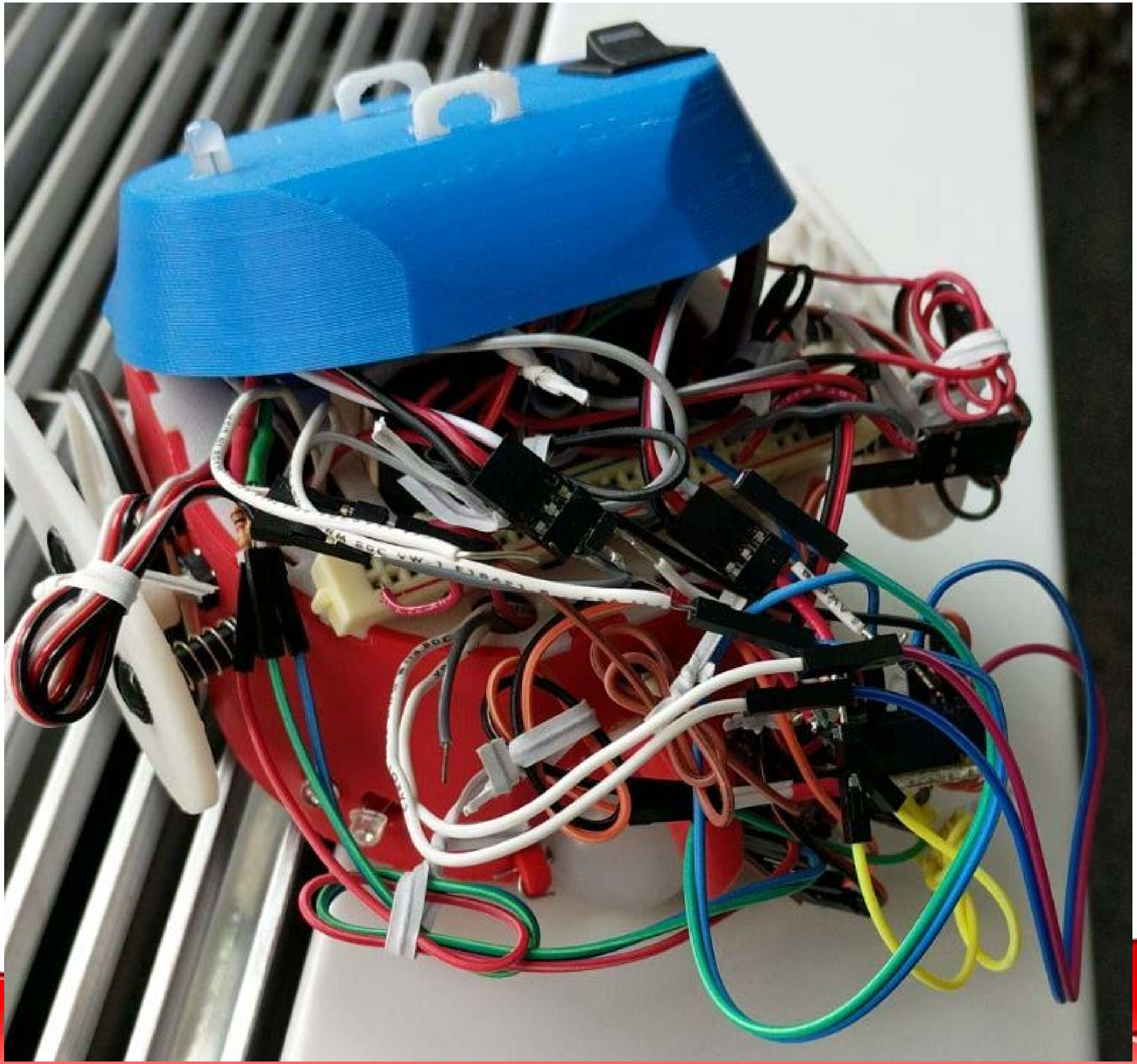
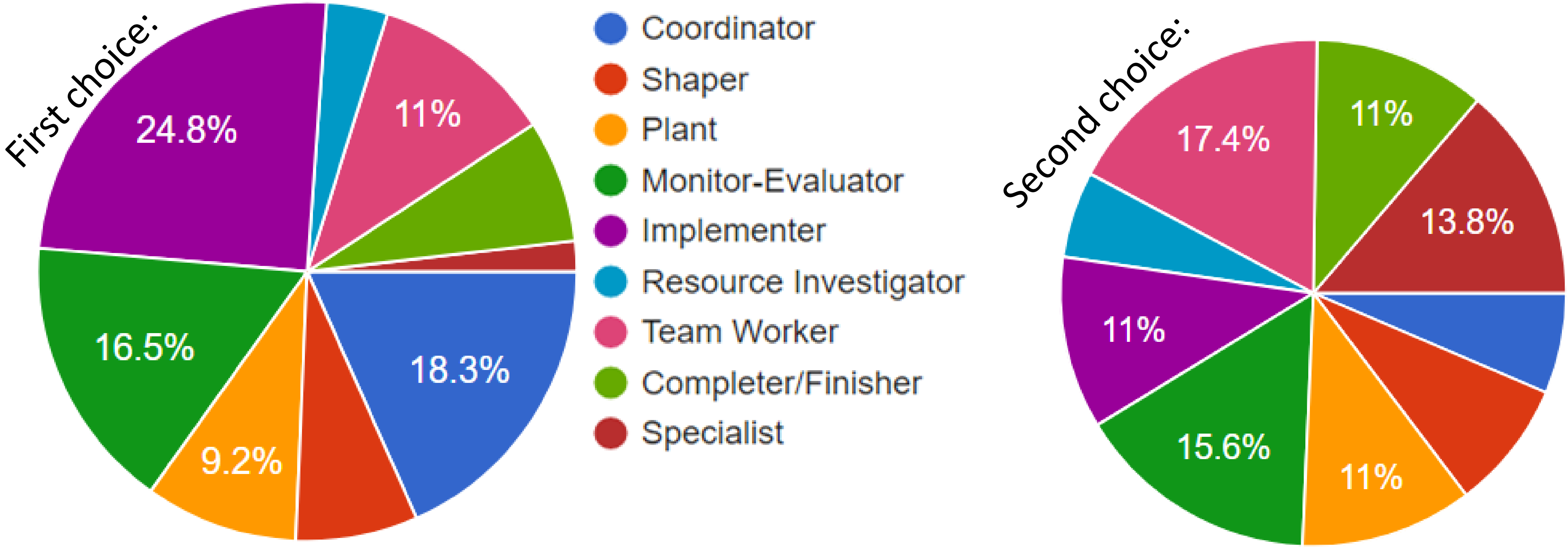
Intelligent Physical Systems



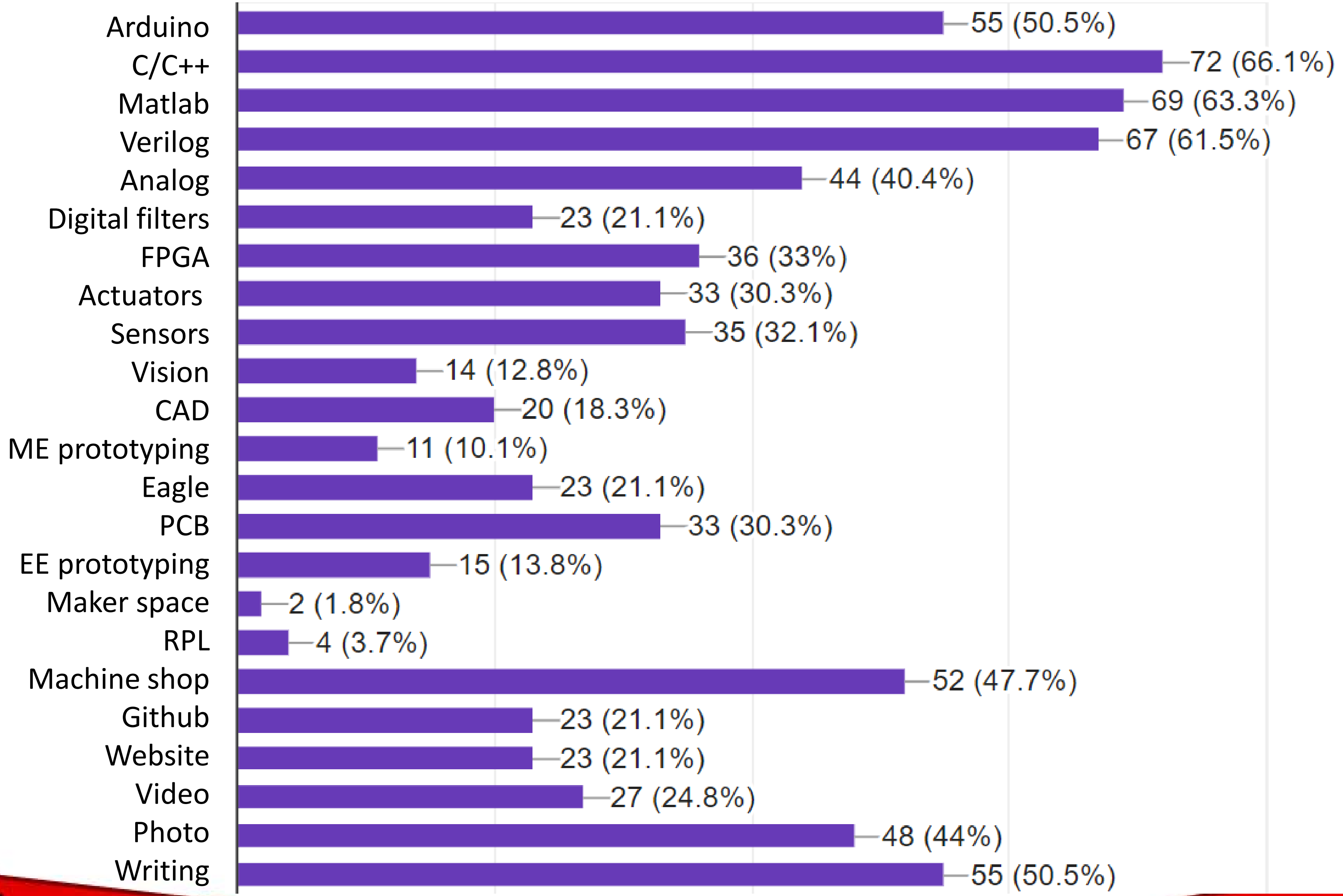
Intelligent Physical Systems

- Team compositions



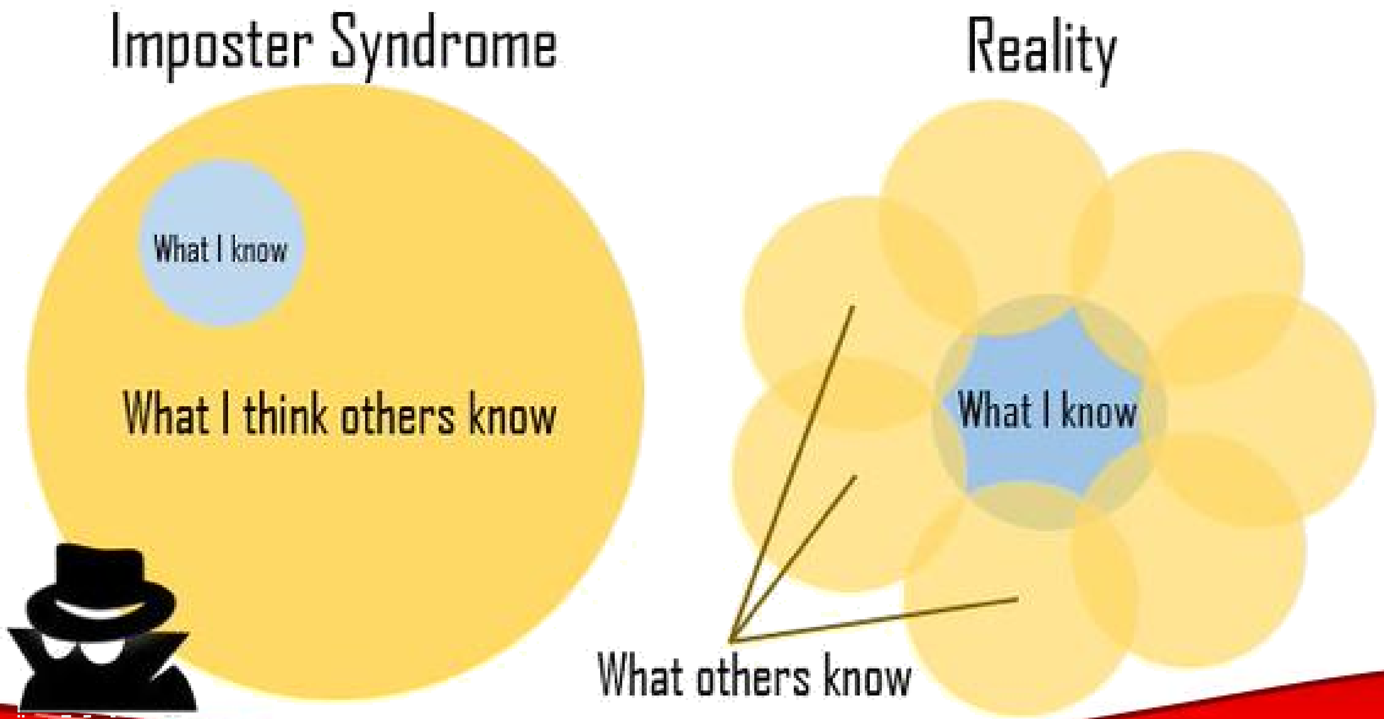
Intelligent Physical Systems

- Team compositions
- Biggest concerns
  - Technical skill
    - Electronics
    - Algorithms
    - Mechanics
    - Presentation



Intelligent Physical Systems

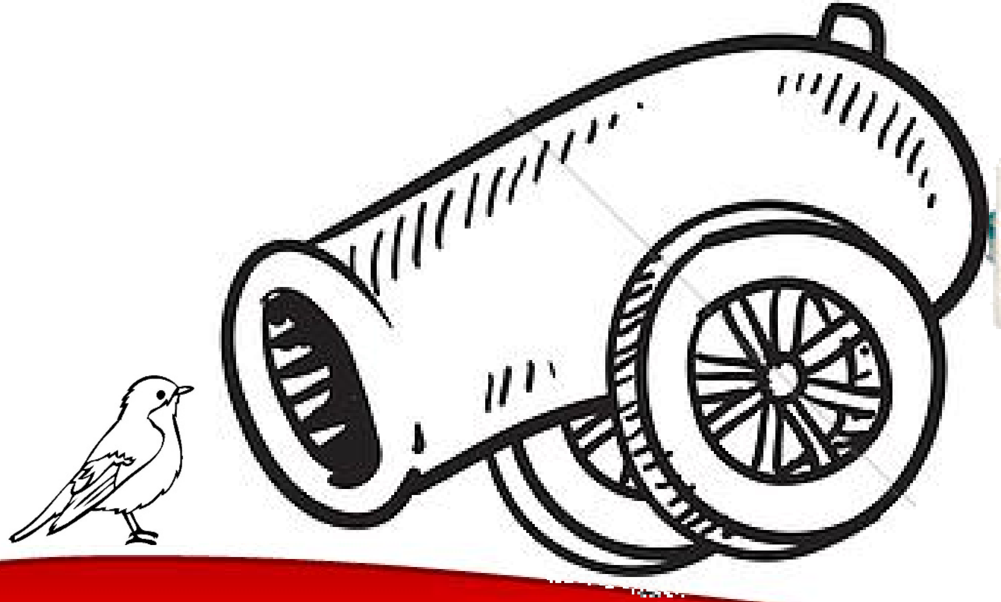
- Team compositions
- Biggest concerns
  - Technical skill (electronics, algorithms, mechanics, presentation)
  - Team
    - Dead weight
    - Not getting to do anything
  - Scheduling / finding time
  - Innovation / creativity
  - Disinterest / laziness

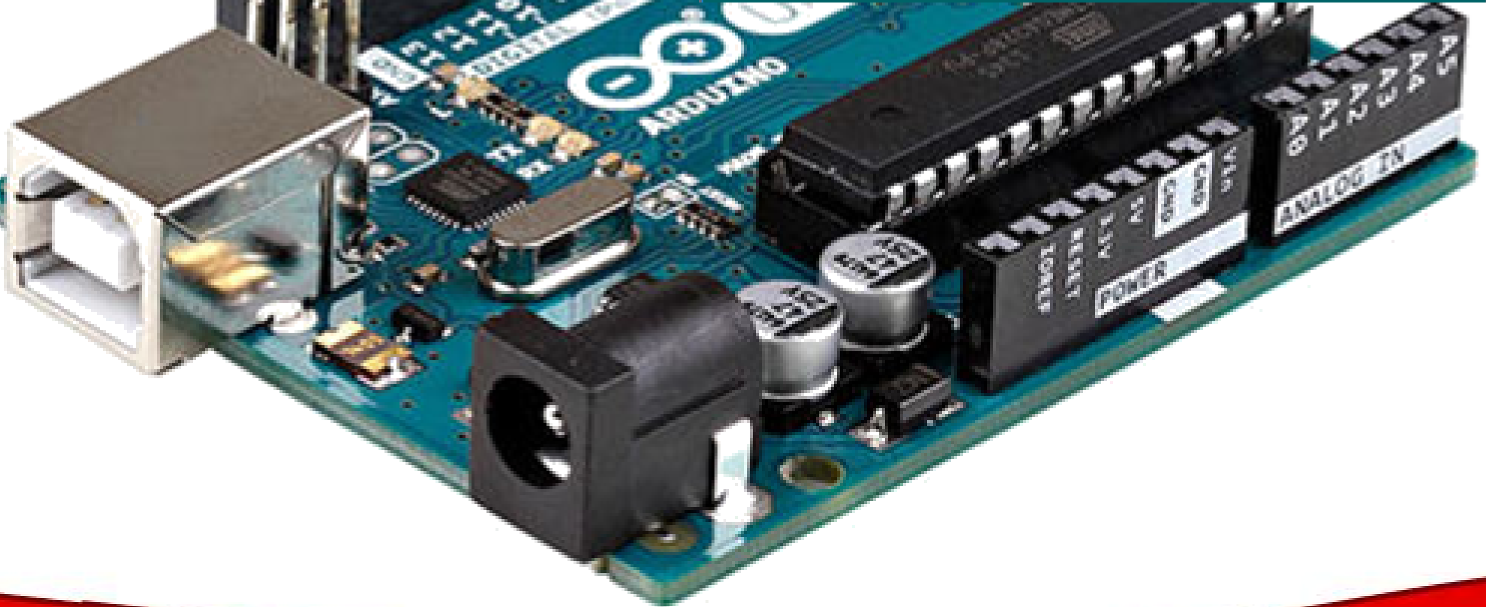


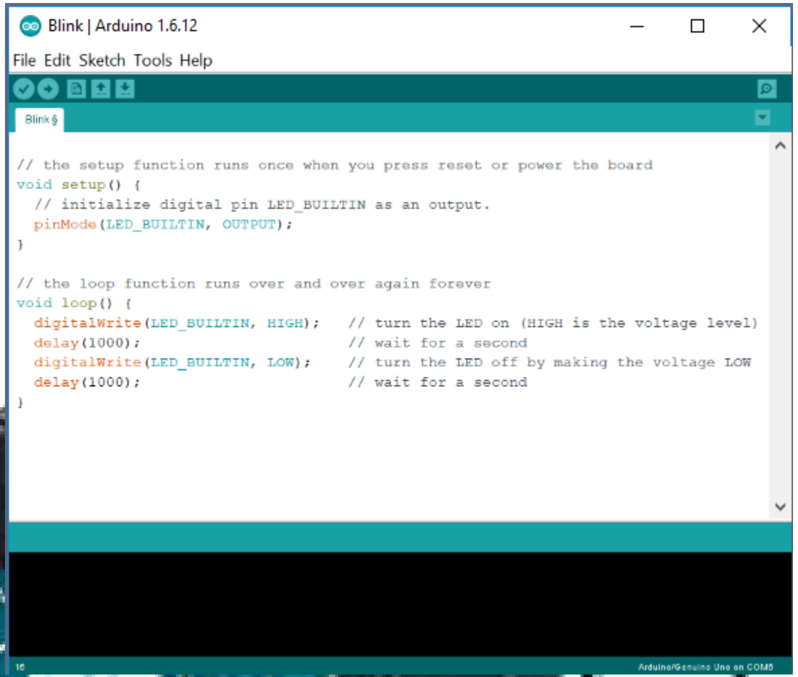
Intelligent Physical Systems

- Introduction
  - What under the hood?
  - How do you program it?

Arduino Uno







```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Alternatives

ARDUINO UNO

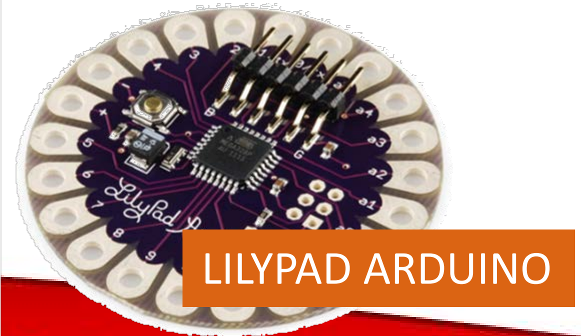
ARDUINO LEONARDO

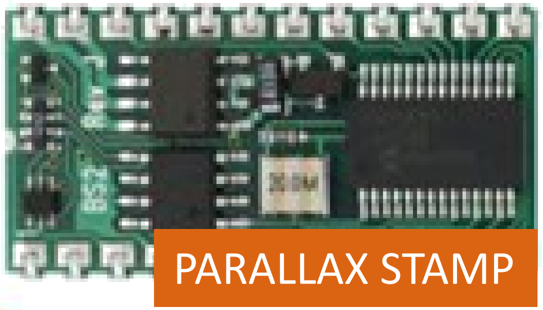
ARDUINO 101

ARDUINO ESPLORA

ARDUINO MICRO

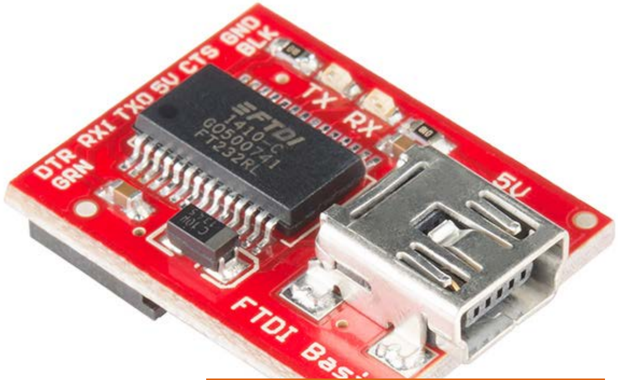
ARDUINO NANO


LILYPAD ARDUINO


PARALLAX STAMP

ARDUINO MINI

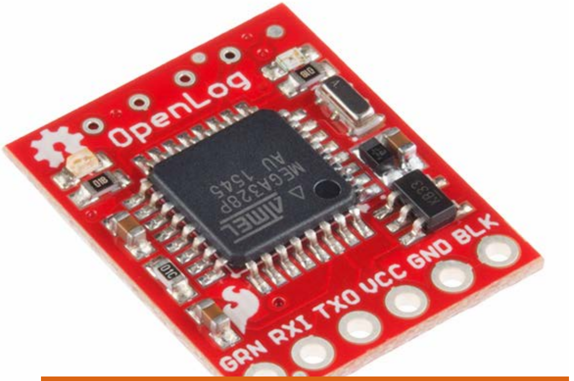
Add-Ons



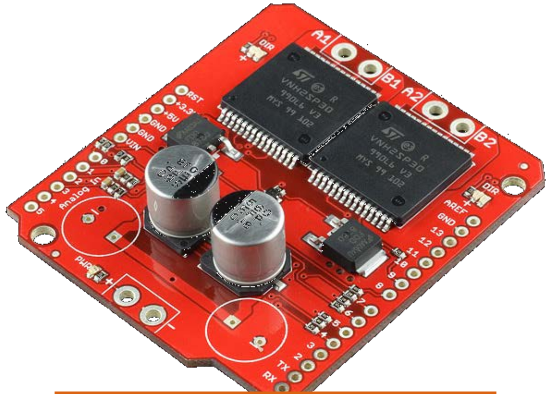
FTDI CONVERTERS

BLUETOOTH

WIRELESS MODULE

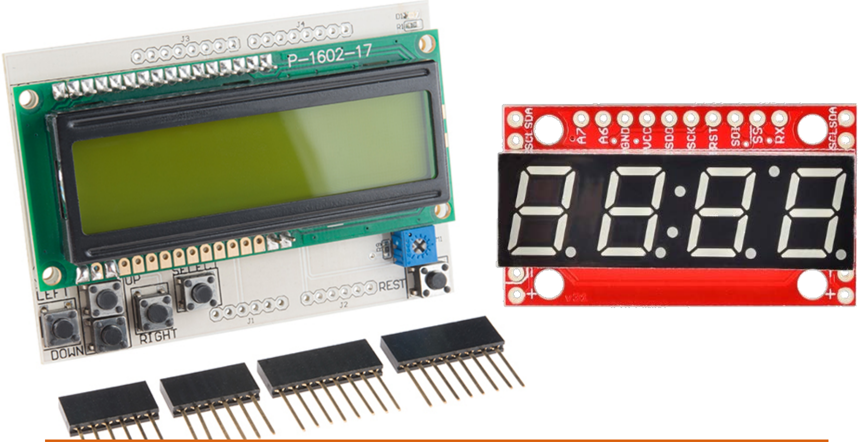


DATA LOGGER (SD CARD)



MOTOR DRIVERS

SENSORS

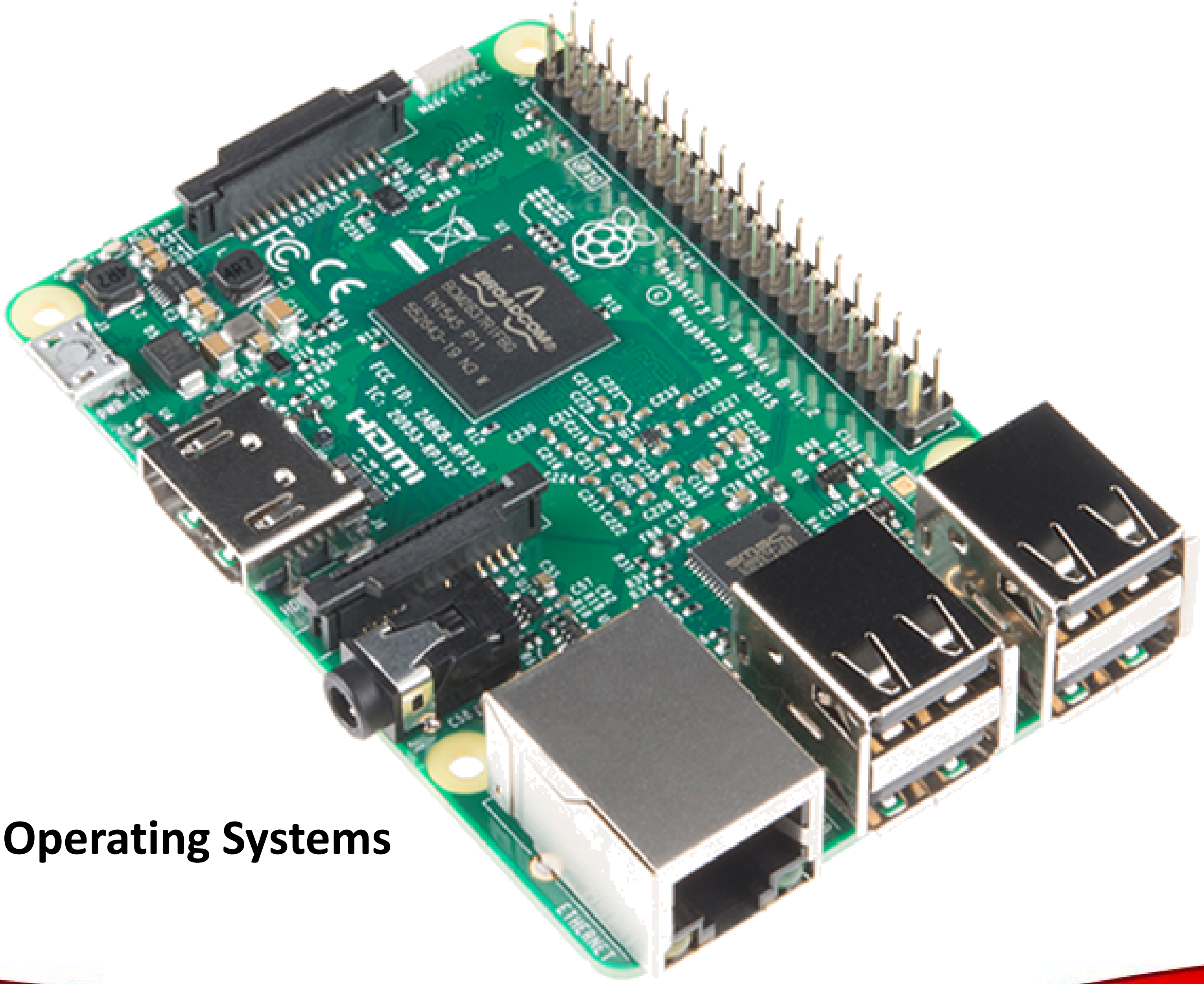


LCD SHIELD / 7-segment LED displays

Alternatives

Raspberry Pi’s and

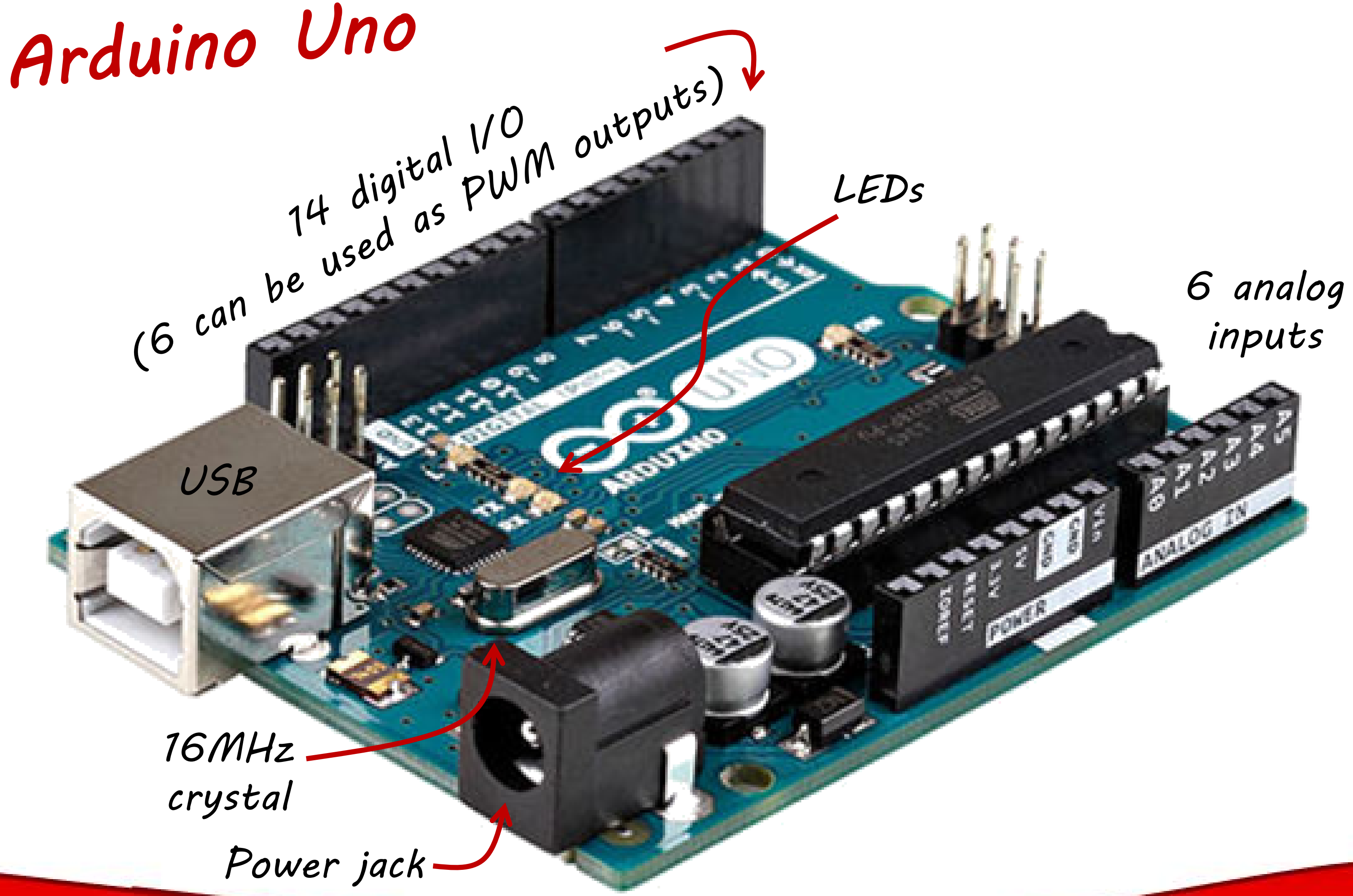
other mini computers



ECE 5725: Design with Embedded Operating Systems

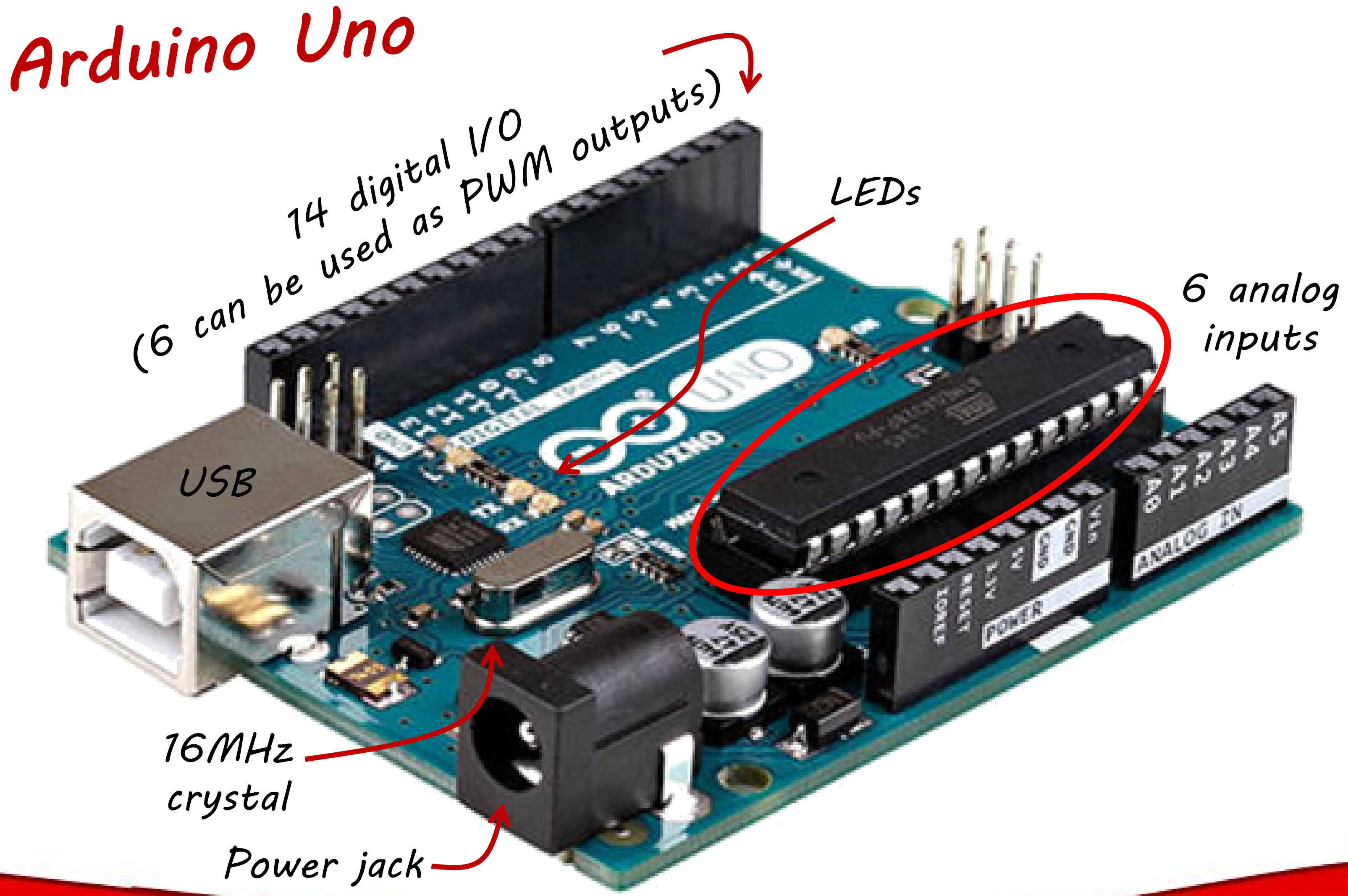
Intelligent Physical Systems

- Introduction
  - What under the hood?
  - How do you program it?



Intelligent Physical Systems

- Microcontroller (ATmega328)



Arduino Uno

14 digital I/O
(6 can be used as PWM outputs)

LEDs

6 analog inputs

USB

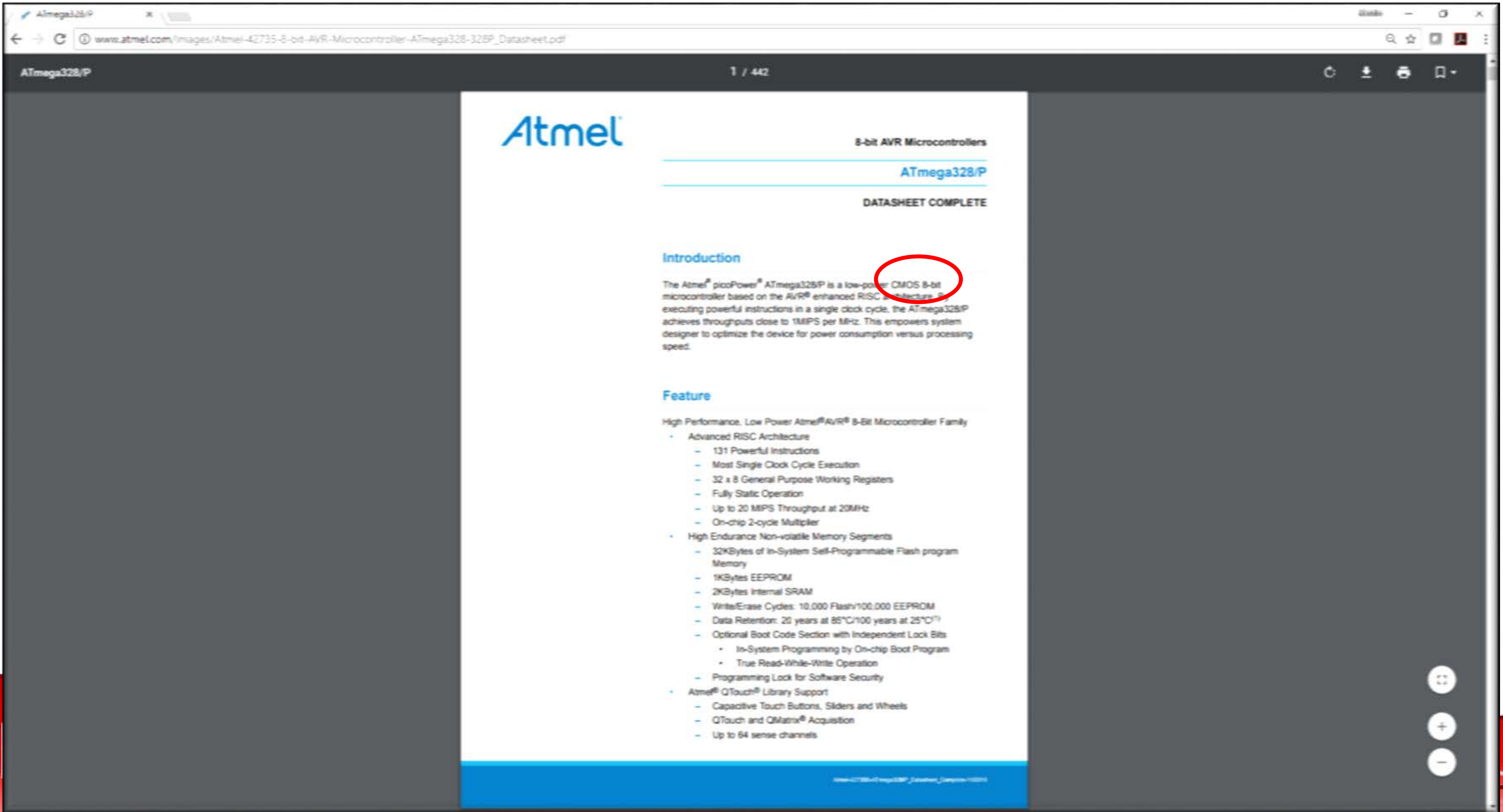
16MHz crystal

Power jack

ATmega328

(Arduino Uno Microcontroller)

http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf


ATmega328/P
www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
ATmega328/P
1 / 442

Atmel

8-bit AVR Microcontrollers

ATmega328/P

DATASHEET COMPLETE

Introduction

The Atmel® picoPower® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

Feature

High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family

- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 32KBytes of In-System Self-Programmable Flash program Memory
  - 1KBytes EEPROM
  - 2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data Retention: 20 years at 85°C/100 years at 25°C[1]
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel® QTouch® Library Support
  - Capacitive Touch Buttons, Sliders and Wheels
  - QTouch and QMatrix® Acquisition
  - Up to 64 sense channels

8 bit Microcontroller

- *What does this mean?*
  - The controller works on 8 bits at a time
- *How many bits is the processor on your computer?*
  - Typically 32 or 64 bit
- *How many bits does a "char" store as?*
  - Trick question! In an 8-bit architecture: a char stores as 1 byte
- *How many bits does a "Boolean" store as?*
  - 1 byte

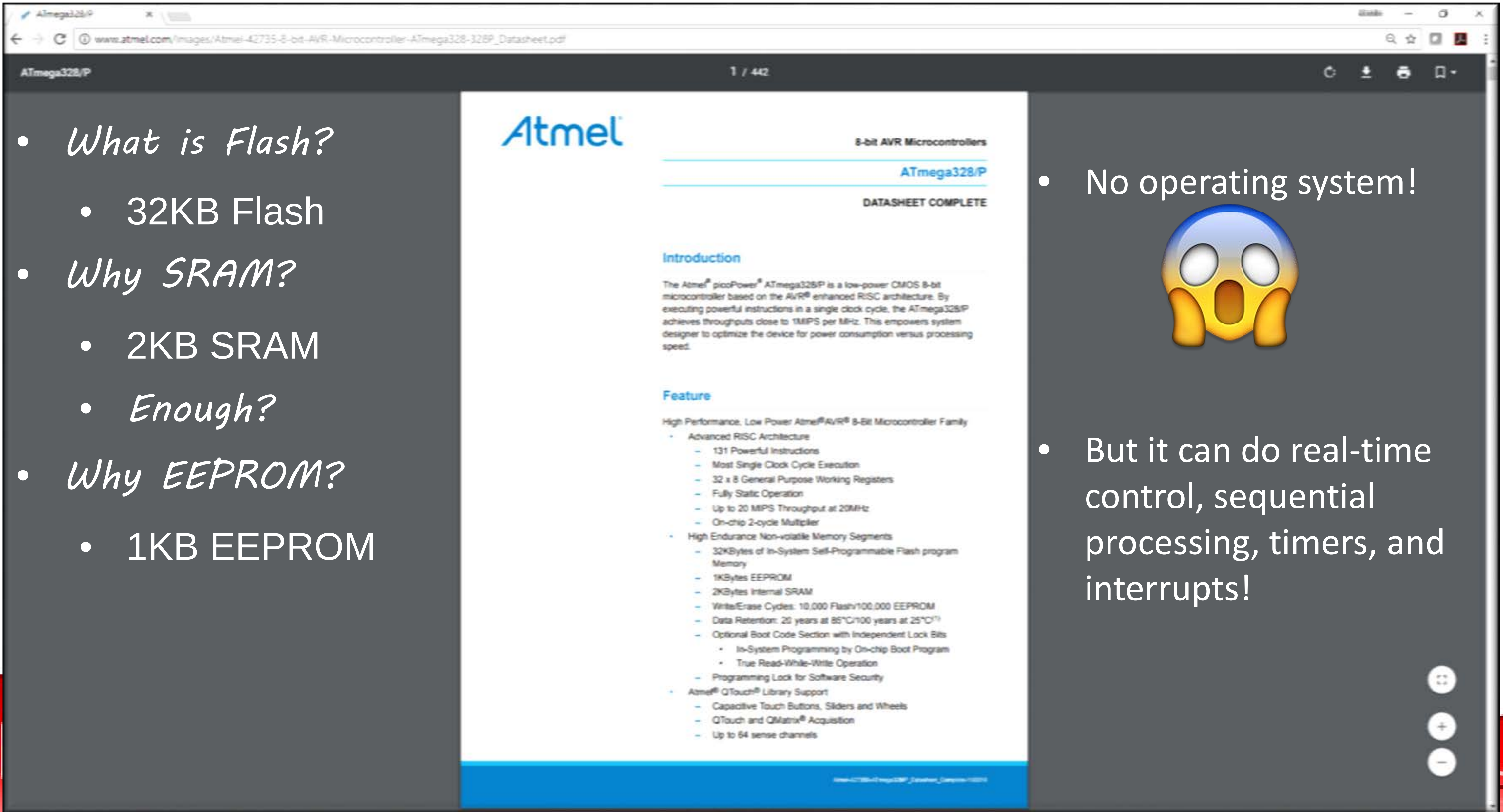
8 bit Microcontroller

- *What does this mean?*
  - The controller works on 8 bits at a time
- 8 bit controllers are suitable for *low power, less compute intensive* applications
- BUT be mindful:
  - `int variable_name;`
  - *What is the maximum number I can store in an integer?*
  - `32,767 = 2^15-1`
  - *What is the minimum number I can store in an integer?*
  - `-32,768`
  - *What happens if I store something larger than 2^{15}-1?*
  - `Overflow!`
  - *What other variable can I use if I want something bigger than 2^{15}-1?*
  - `Unsigned int, long, double, float`

ATmega328

(Arduino Uno Microcontroller)

http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf



ATmega328/P
www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
ATmega328/P
1 / 442

- *What is Flash?*
  - 32KB Flash
- *Why SRAM?*
  - 2KB SRAM
  - *Enough?*
- *Why EEPROM?*
  - 1KB EEPROM

Atmel

8-bit AVR Microcontrollers

ATmega328/P

DATASHEET COMPLETE

Introduction

The Atmel® picoPower® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

Feature

High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family

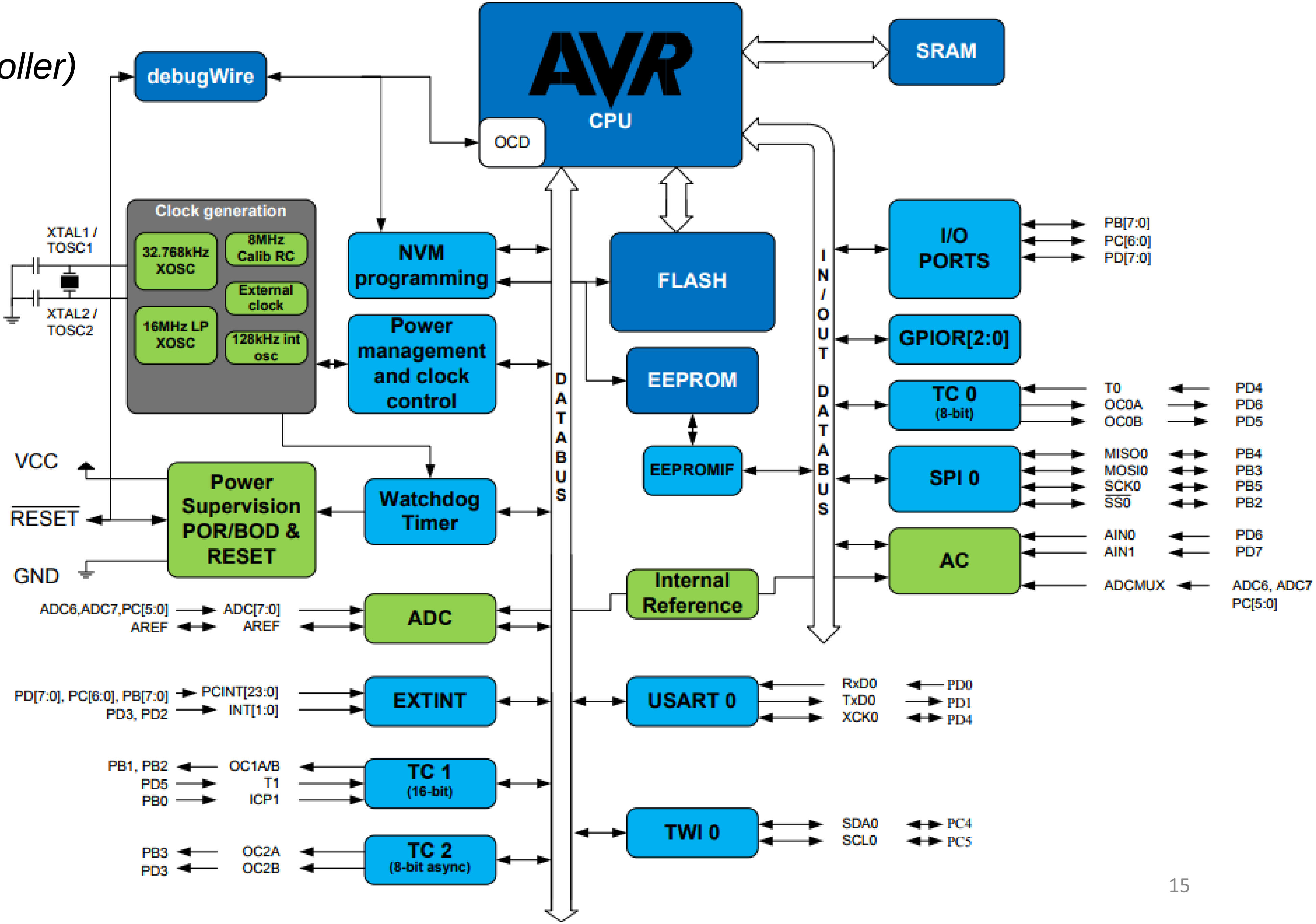
- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 32KBytes of In-System Self-Programmable Flash program Memory
  - 1KBytes EEPROM
  - 2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data Retention: 20 years at 85°C/100 years at 25°C[1]
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel® QTouch® Library Support
  - Capacitive Touch Buttons, Sliders and Wheels
  - QTouch and QMatrix® Acquisition
  - Up to 64 sense channels

- No operating system! 😱
- But it can do real-time control, sequential processing, timers, and interrupts!

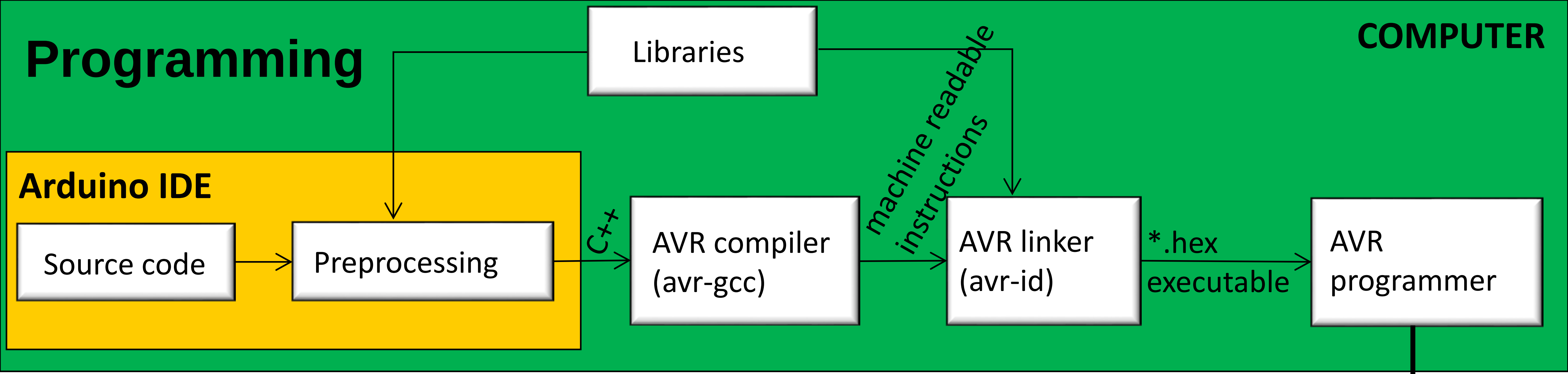
ATmega328

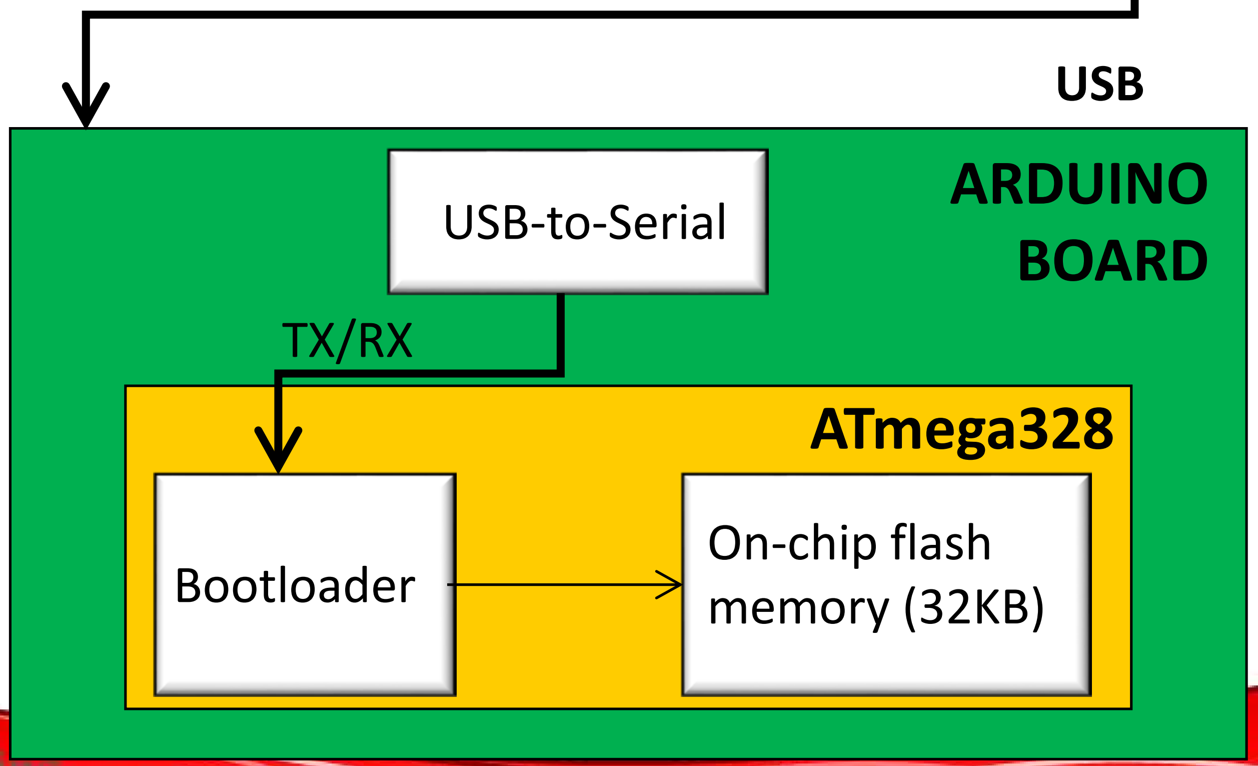
(Arduino Uno Microcontroller)

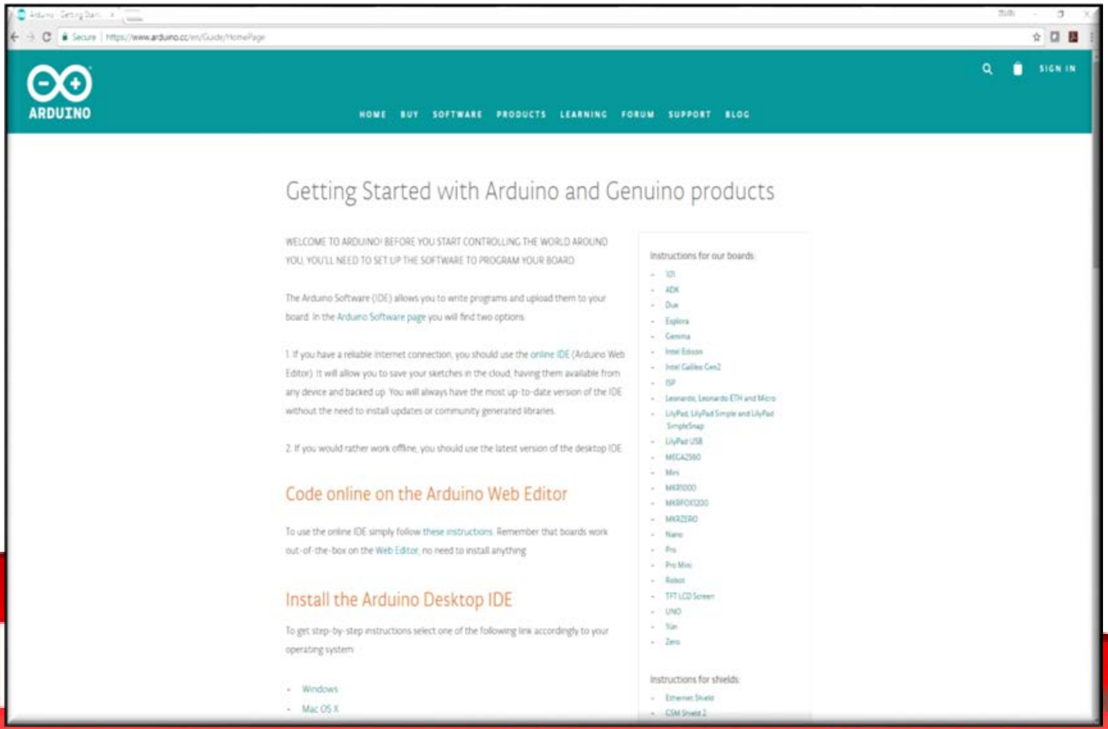
- CPU
- Memory
- Clock
- Fuses
  - Clock speed
  - Brownout
  - Watchdog
  - etc.
- ADC
- Timer/counters
- Interrupts
- SPI/TWI/USART
- I/O ports



Programming






Arduino - Getting Start...
Secure | https://www.arduino.cc/en/Guide/HomePage
ARDUINO
SIGN IN
HOME BUY SOFTWARE PRODUCTS LEARNING FORUM SUPPORT BLOG

Getting Started with Arduino and Genuino products

WELCOME TO ARDUINO! BEFORE YOU START CONTROLLING THE WORLD AROUND YOU, YOU'LL NEED TO SET UP THE SOFTWARE TO PROGRAM YOUR BOARD

The Arduino Software (IDE) allows you to write programs and upload them to your board. In the Arduino Software page you will find two options:

1. If you have a reliable Internet connection, you should use the online IDE (Arduino Web Editor). It will allow you to save your sketches in the cloud, having them available from any device and backed up. You will always have the most up-to-date version of the IDE without the need to install updates or community generated libraries.

2. If you would rather work offline, you should use the latest version of the desktop IDE.

Code online on the Arduino Web Editor

To use the online IDE simply follow these instructions. Remember that boards work out-of-the-box on the Web Editor, no need to install anything.

Install the Arduino Desktop IDE

To get step-by-step instructions select one of the following link accordingly to your operating system.

- Windows
- Mac OS X

Instructions for our boards:

- 101
- ADK
- Due
- Esplora
- Gemma
- Intel Edison
- Intel Galileo Gen2
- ISP
- Leonardo, Leonardo ETH and Micro
- LilyPad, LilyPad Simple and LilyPad SimpleSnap
- LilyPad USB
- MEGA2560
- Mini
- MKR1000
- MKRFOX1200
- MKRZERO
- Nano
- Pro
- Pro Mini
- Robot
- TFT LCD Screen
- UNO
- Yun
- Zero

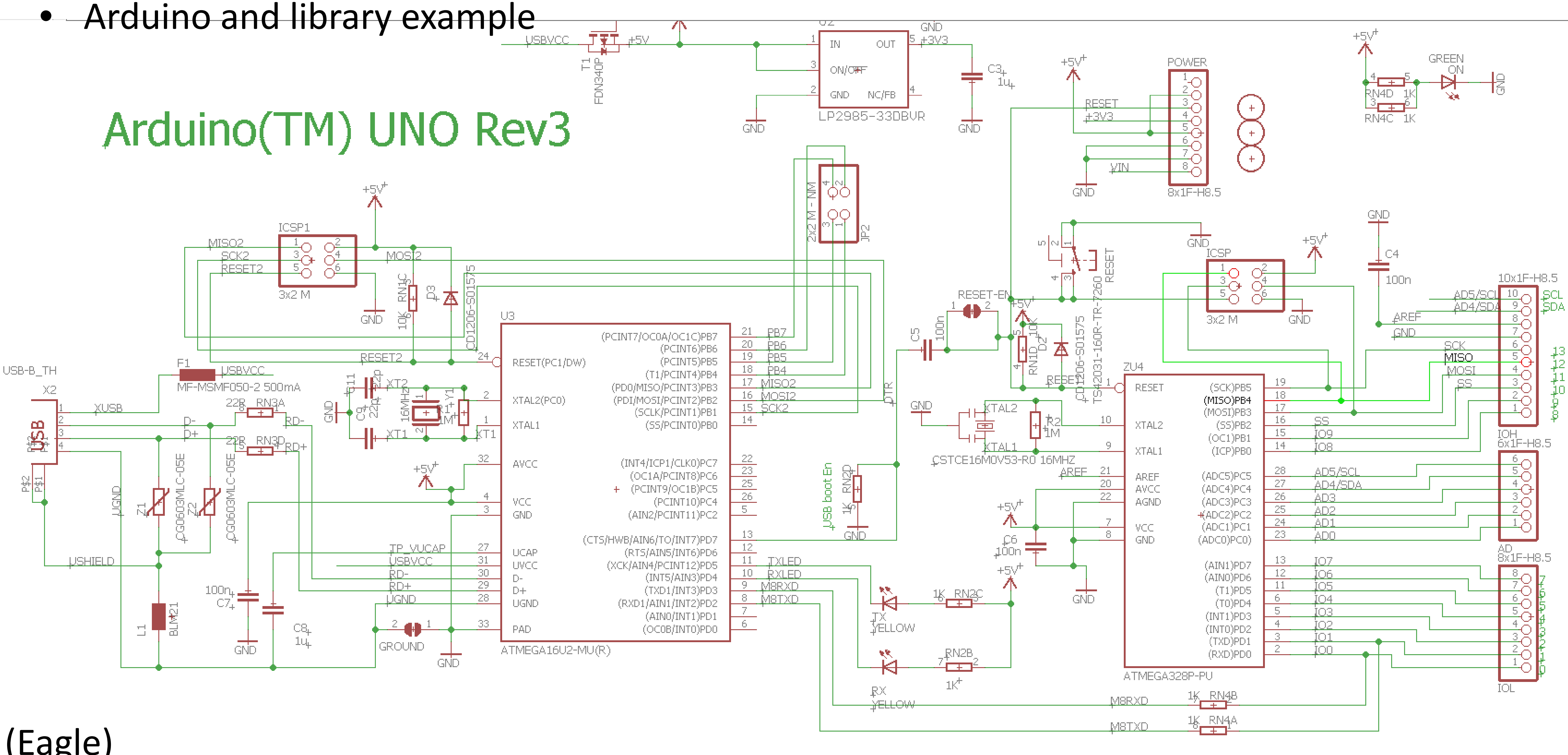
Instructions for shields:

- Ethernet Shield
- GSM Shield 2

Arduino and Libraries

https://store.arduino.cc/usa/arduino-uno-rev3

- Arduino and library example



(Eagle)

Arduino and Libraries

https://store.arduino.cc/usa/arduino-uno-rev3

- Arduino and library example
- How does the Arduino IDE know what digitalWrite() /digitalRead() means?
  - wiring_digital.c library

```
void digitalWrite(uint8_t pin, uint8_t val)
    uint8_t timer = digitalPinToTimer(pin);
    uint8_t bit = digitalPinToBitMask(pin);
    uint8_t port = digitalPinToPort(pin);
    volatile uint8_t *out;
    if (port == NOT_A_PIN) return; // If PWM output, turn it off before DigWrite
    if (timer != NOT_ON_TIMER)
        turnOffPWM(timer);

    out = portOutputRegister(port);
    uint8_t oldSREG = SREG;
    cli();

    if (val == LOW)
        *out &= ~bit;
    else
        *out |= bit;
    SREG = oldSREG;
```

Arduino and Libraries

https://store.arduino.cc/usa/arduino-uno-rev3

- Arduino and library example
- How does the AVR compiler know what digitalWrite() /digitalRead() means?
  - wiring_digital.c library
- How does the AVR compiler know what the LED_BUILTIN is?
  - pins_arduino.h library

Arduino and Libraries

https://store.arduino.cc/usa/arduino-uno-rev3

- Arduino and library example
- How does the AVR compiler know what digitalWrite() /digitalRead() means?
  - wiring_digital.c library
- How does the AVR compiler know what the LED_BUILTIN is?
  - pins_arduino.h library
- How does the AVR compiler know what pin output/inputs are?
  - iom328p.h library

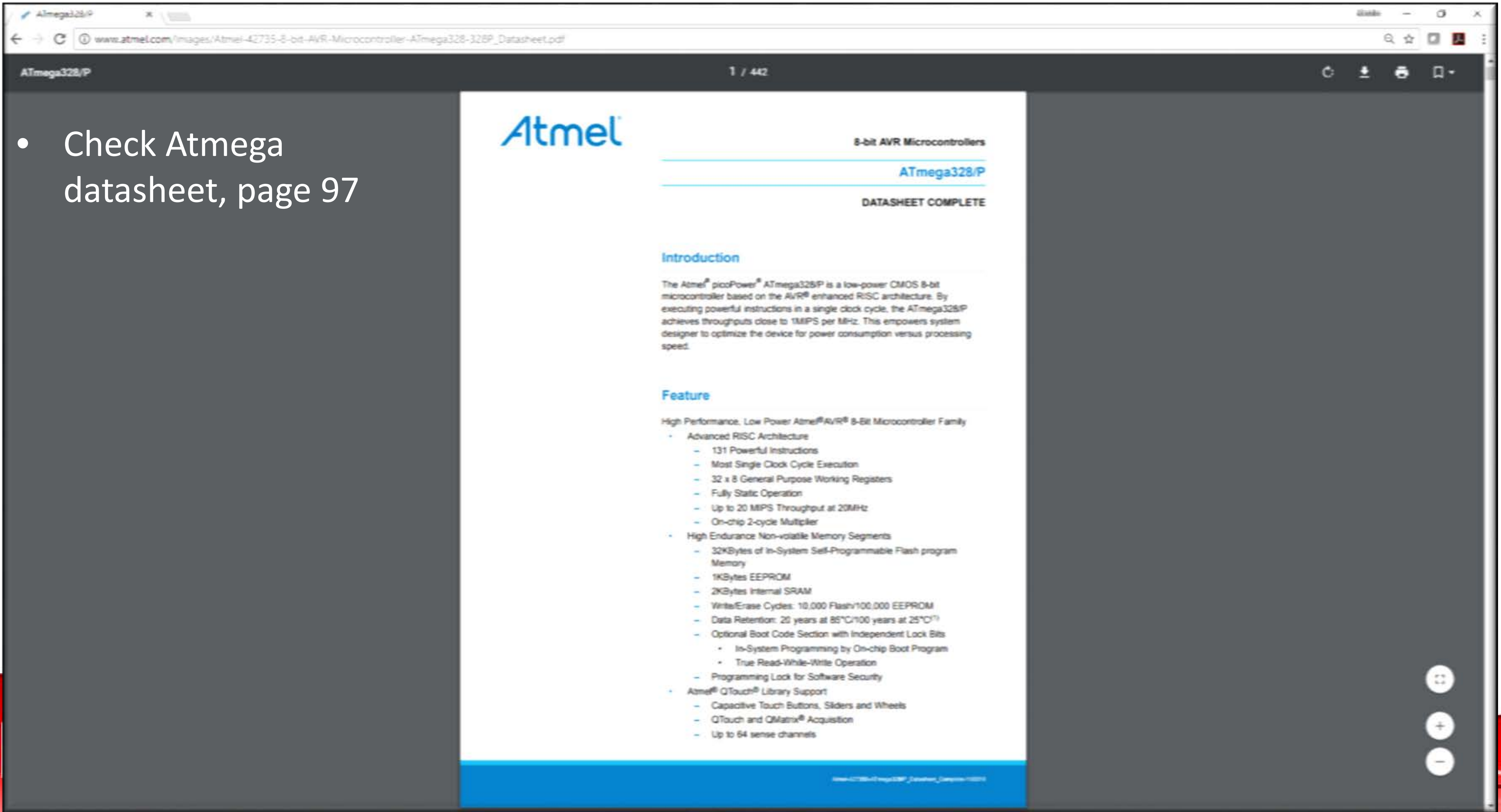
processor name

ATmega328

(Arduino Uno Microcontroller)

http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

- Check Atmega datasheet, page 97



ATmega328/P
www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
ATmega328/P
1 / 442
Atmel

8-bit AVR Microcontrollers

ATmega328/P

DATASHEET COMPLETE

Introduction

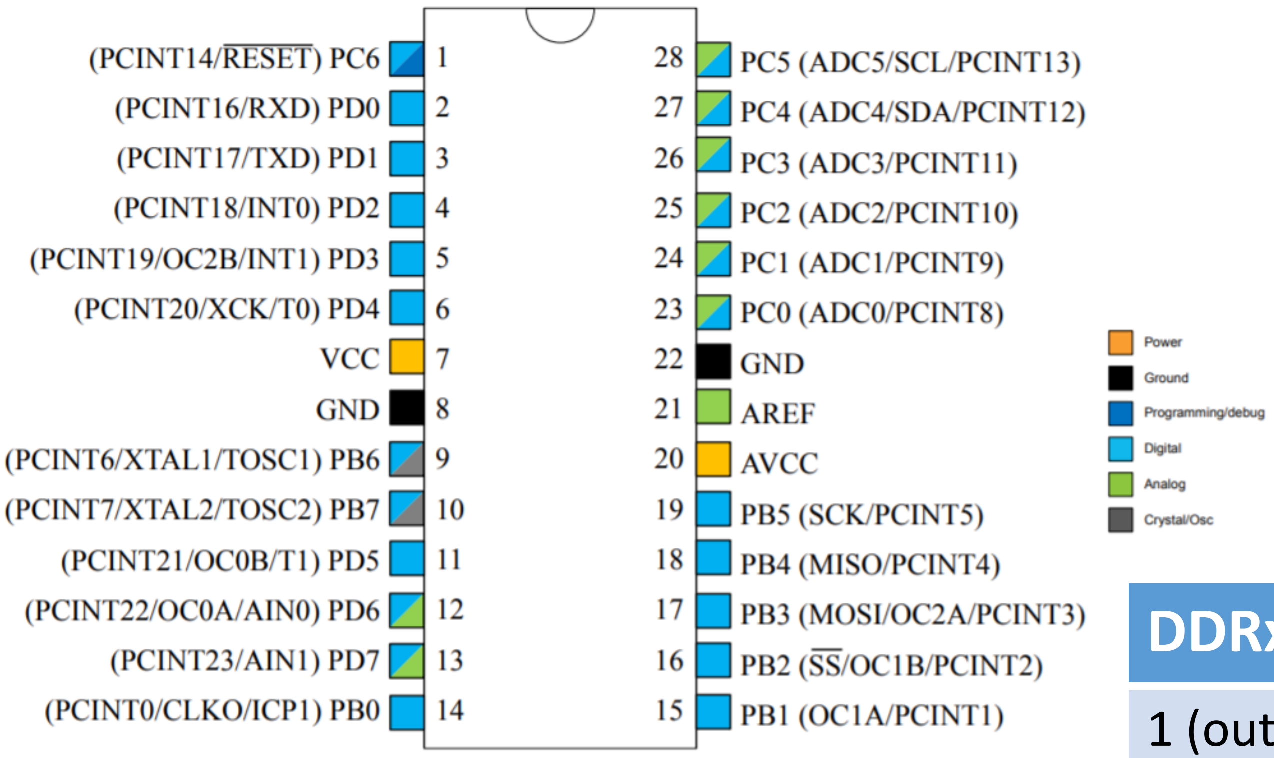
The Atmel® picoPower® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

Feature

High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller Family

- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 32KBytes of In-System Self-Programmable Flash program Memory
  - 1KBytes EEPROM
  - 2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data Retention: 20 years at 85°C/100 years at 25°C[1]
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel® QTouch® Library Support
  - Capacitive Touch Buttons, Sliders and Wheels
  - QTouch and QMatrix® Acquisition
  - Up to 64 sense channels

Input/Outputs



- library: iom328p.h

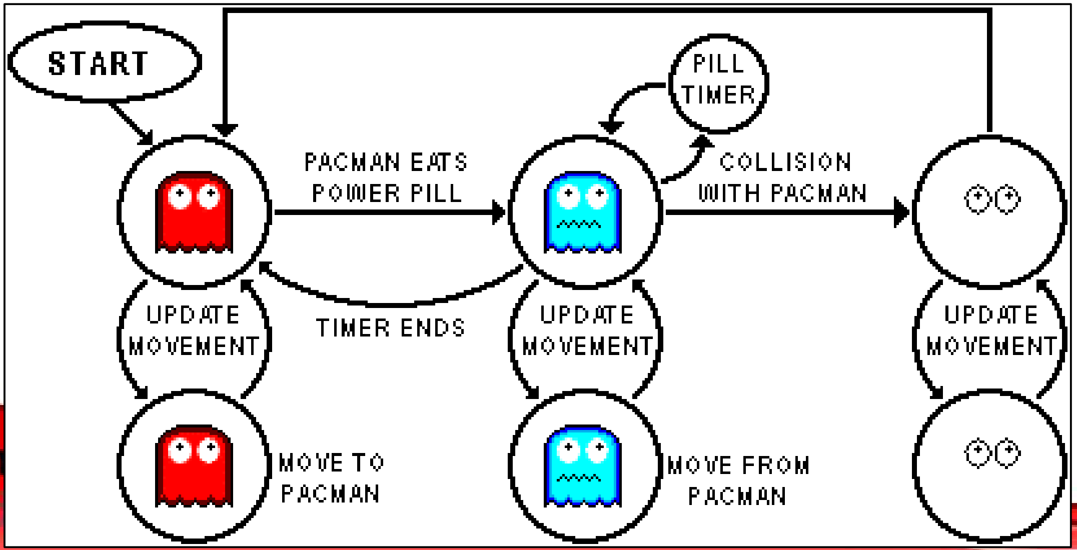
- Max 10mA per pin (20mA per port)
- Why tri-state?
- Why pull-up?

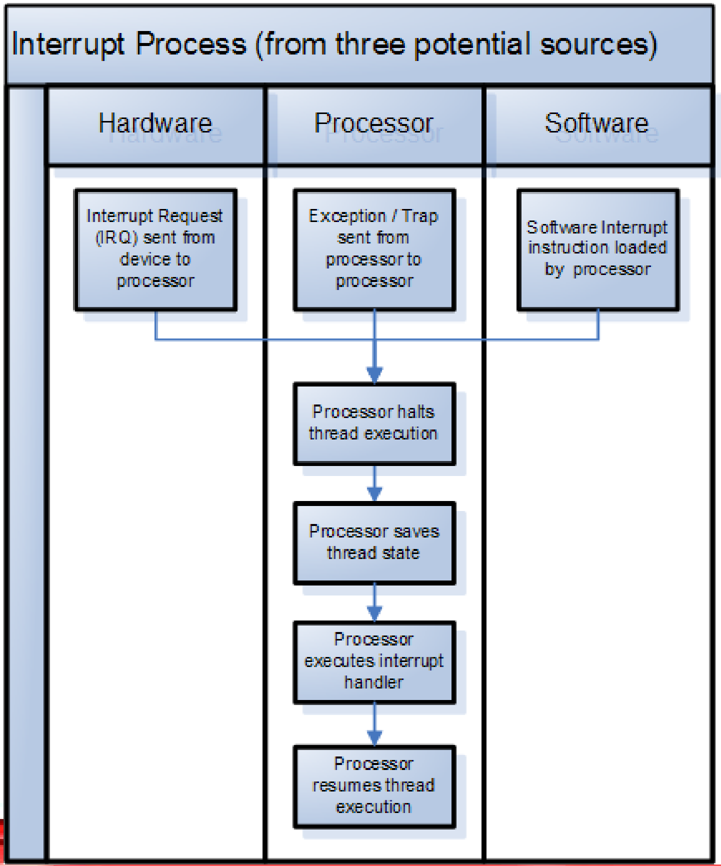
DDRxn	PORTxn	PINxn	Setup
1 (output)	0	X	Output low
1 (output)	1	X	Output high
1 (output)	X	1	Toggle output
0 (input)	-	X	Input (tri-state)
0 (input)	1	X	Pull-up

What to do without an Operating System?

- Loop
  - Execution time depends on instructions in the loop
- Round-Robin
  - Every process is allotted the same time (some may take many rounds to compute)
- Event/interrupt driven
  - Guaranteed processing within some defined time slot
- Combination

State machines are very helpful!




Interrupt Process (from three potential sources)
Hardware
Processor
Software
Interrupt Request (IRQ) sent from device to processor
Exception / Trap sent from processor to processor
Software Interrupt instruction loaded by processor
Processor halts thread execution
Processor saves thread state
Processor executes interrupt handler
Processor resumes thread execution

Lab Logistics

- Lab desks
  - Two per team
  - …Crowding
  - Cleanliness
  - Mazes
- Boxes

- Show up to *every* lab session (and as many open lab hours as needed)
- Aug 31st, Friday: 1) Team contract, 2) team website, 3) add a link to your contract, and 4) send your website link to Kirstin.
- Lab 1 will be graded in two weeks (Sept 7th)

*Monday labs → Open lab hours

Go Build Robots!

Class website: https://cei-lab.github.io/ece3400-2018/

Piazza: https://piazza.com/cornell/fall2018/ece3400/home