Microcontroller

Course Teacher:

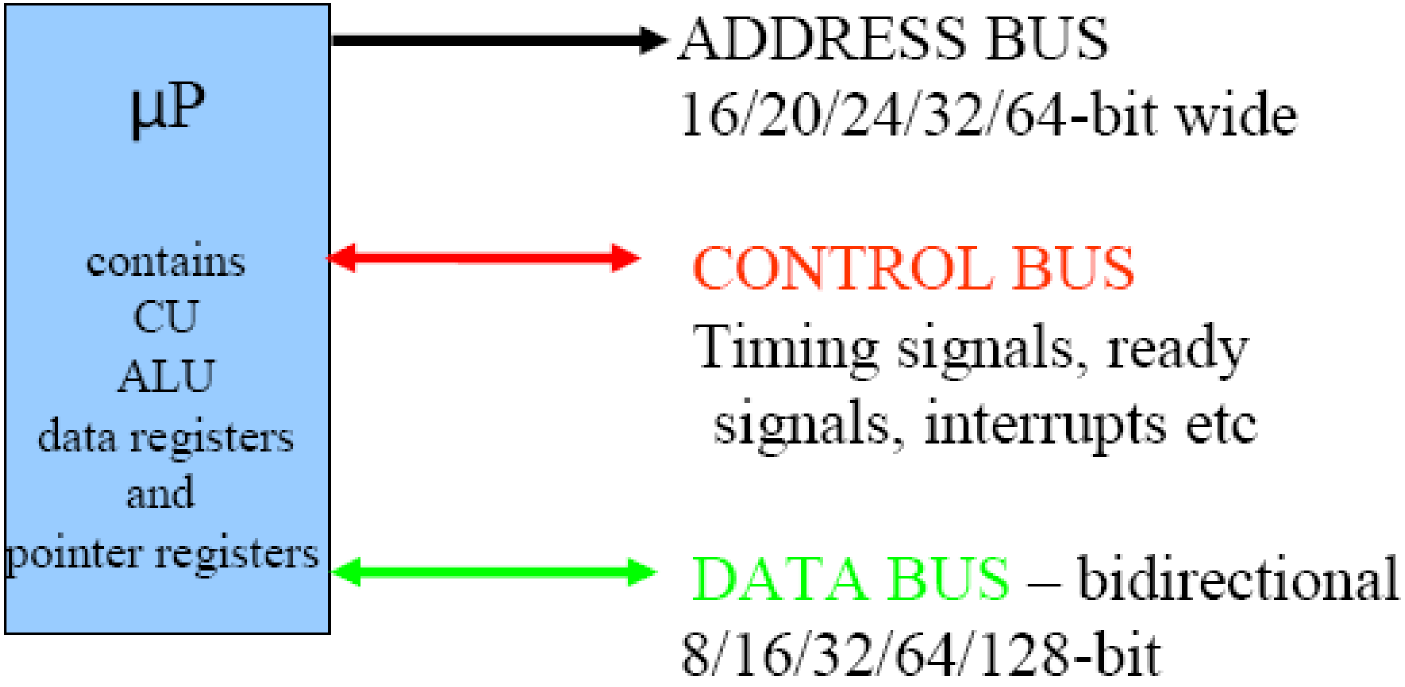
Md. Obaidur Rahman, Ph.D.
Professor
Department of Computer Science and Engineering (CSE),
Dhaka University of Engineering & Technology (DUET), Gazipur.

Course ID: CSE - 4619
Course Title: Peripherals, Interfacing and Embedded Systems
Department of Computer Science and Engineering (CSE),
Islamic University of Technology (IUT), Gazipur.

Lecture References:

- **Web Materials:**
  - http://en.wikipedia.org/wiki/Atmel_AVR
  - www.atmel.com
  - http://www.atmel.com/Images/2466S.pdf
  - www.microchip.com
  - www.alldatasheet.com
  - www.avrfreaks.net
  - www.arduino.cc

Microprocessor – Basic Concept



μP

contains
CU
ALU
data registers
and
pointer registers

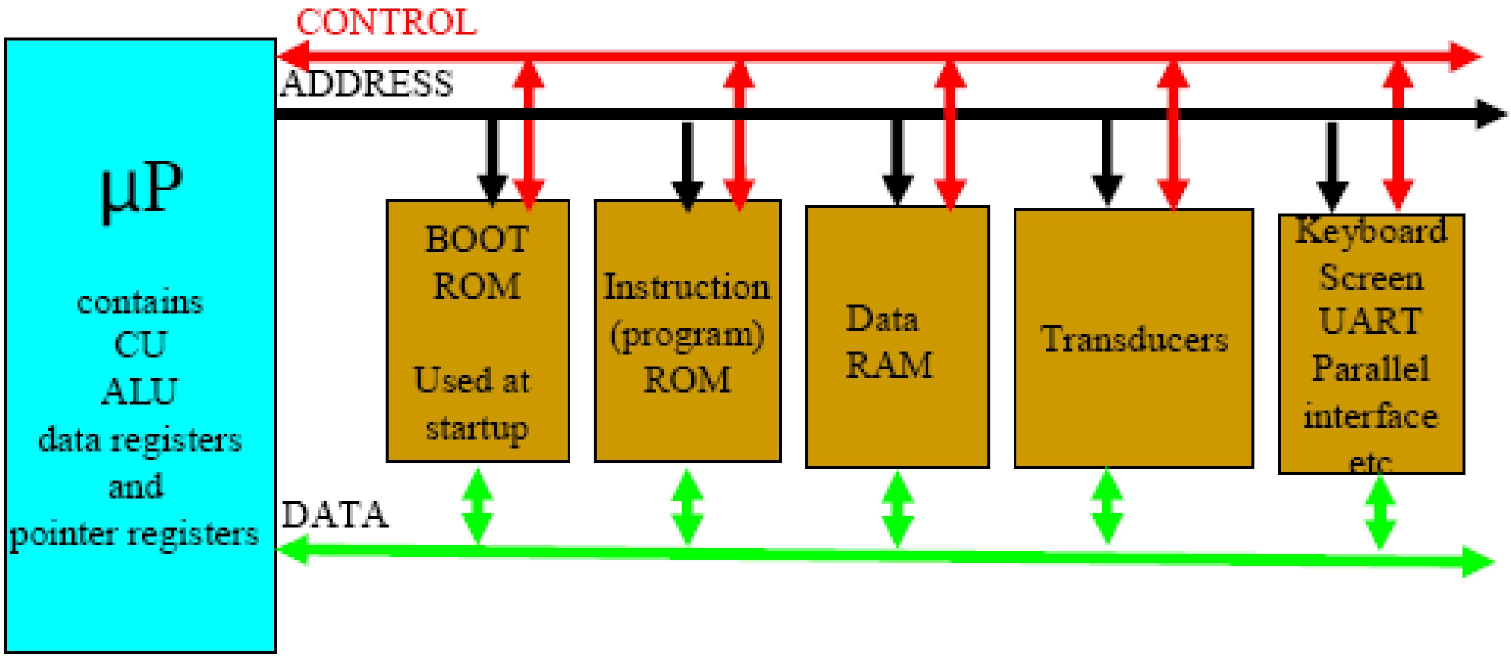
ADDRESS BUS
16/20/24/32/64-bit wide

CONTROL BUS
Timing signals, ready signals, interrupts etc

DATA BUS – bidirectional
8/16/32/64/128-bit

Microprocessor – Basic Concept

- Microprocessor by itself is completely useless.
  - MP must have external peripherals to interact with outside world.



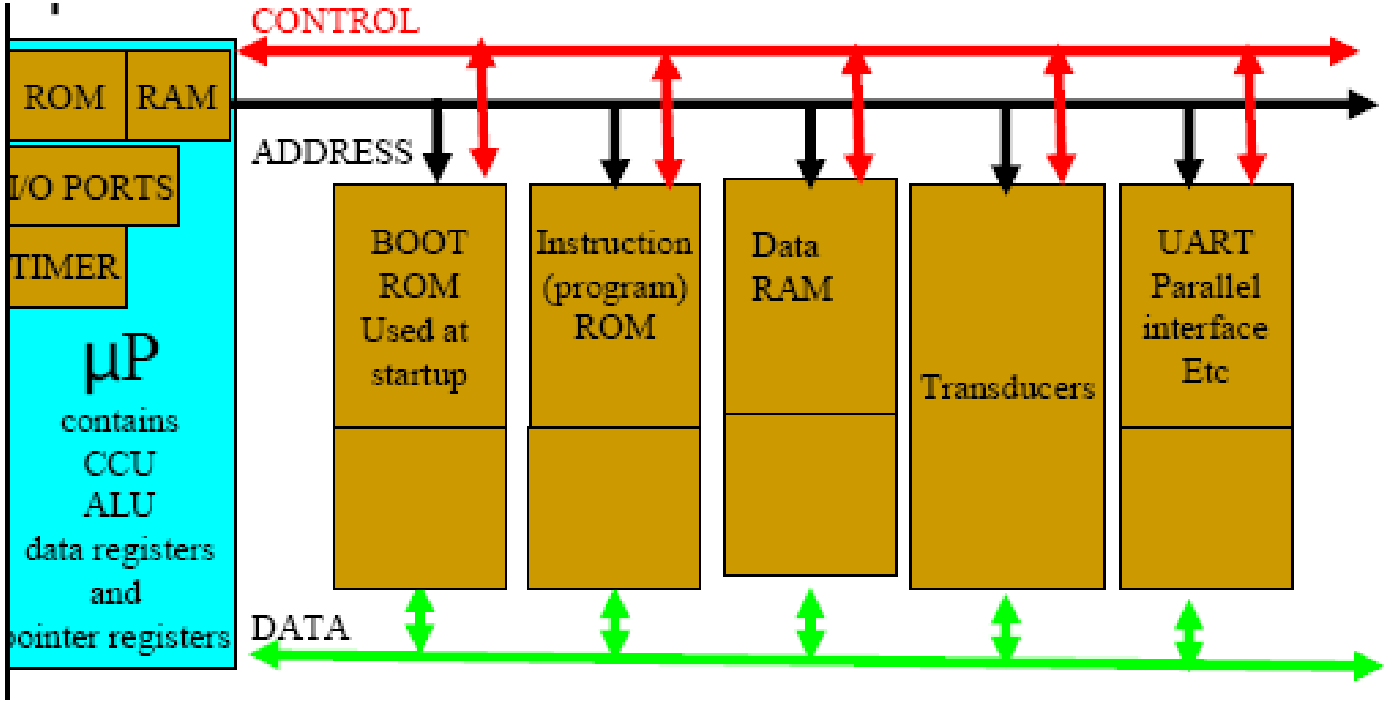
Many chips on mother board

Issues and Challenges for MP

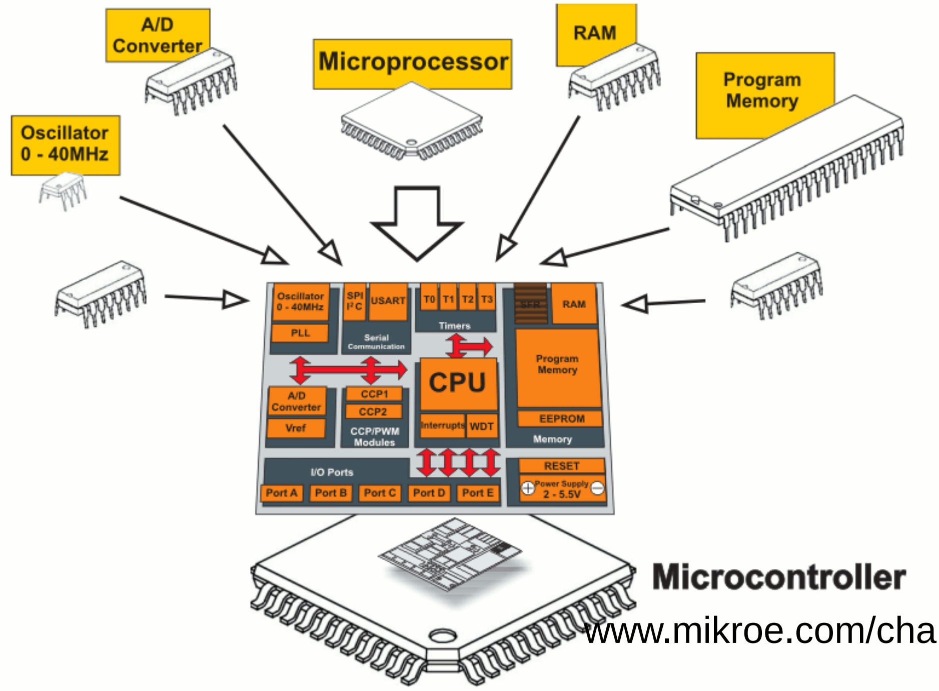
- Issues with increasing number of external devices or chips.
  - Many chips mean **many interconnections**
  - **Many interconnection** means -
    - Mechanical failure rates **increased**
    - Design time **increased**
    - Cost **increased**
    - Board size **increased**
    - Compatibility between parts **increased**

So … Solution is Microcontroller !!

- Microcontroller – put a **limited** amount of most commonly used resources "inside" the chip
  - A "**limited**" amount is often "**enough**" for many applications



What is a Microcontroller



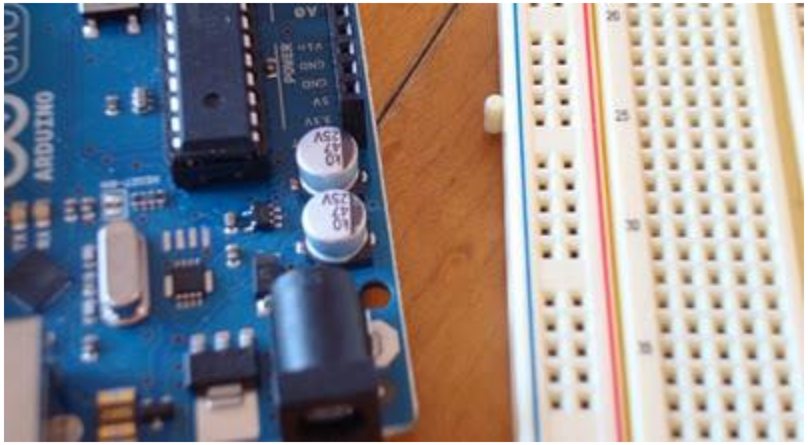
www.mikroe.com/chapters/view/1

Fig. 0-1 Microcontroller versus Microprocessor

- A small computer on a single chip
  - containing a processor, memory, and input/output
- Typically "**embedded**" inside some device that they control
- A microcontroller is often small and low cost

What is a Development Board

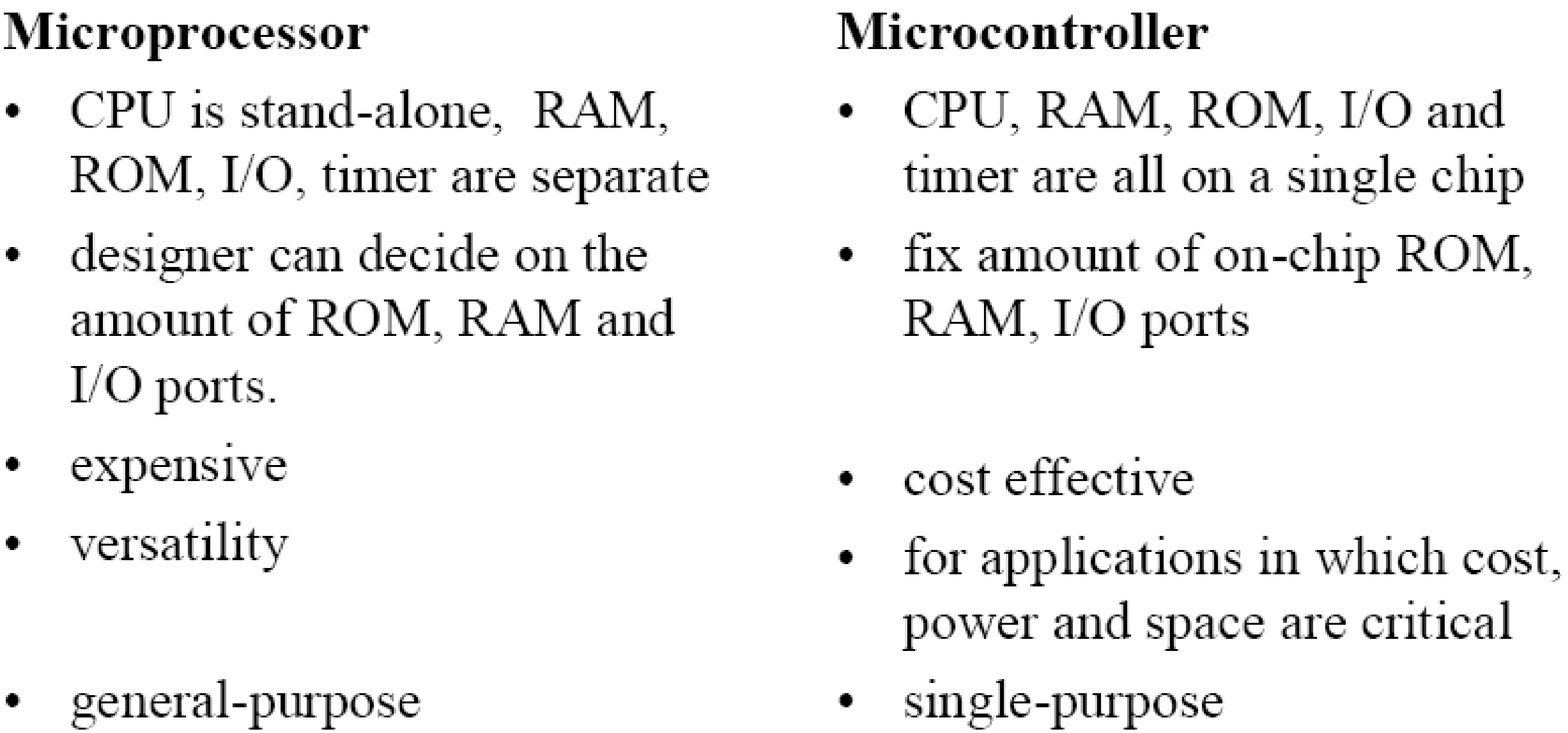
- A printed circuit board designed to facilitate work with a particular microcontroller.
- Typical components include:
  - power circuit
  - programming interface
  - basic input; usually buttons and LEDs
  - I/O pins



Advantages of MC over MP

- Number of Pins count **decreased**
- Design time **decreased**
- Board layout size **decreased**
- Upgrade path **easier** – matching between peripherals for speed
- Cost **decreased** – bulk purchases
- Reliability **increased**
- Common software / hardware design environment **available** from manufacturer

Microprocessor Vs Microcontroller



Microprocessor	Microcontroller
• CPU is stand-alone, RAM, ROM, I/O, timer are separate	• CPU, RAM, ROM, I/O and timer are all on a single chip
• designer can decide on the amount of ROM, RAM and I/O ports.	• fix amount of on-chip ROM, RAM, I/O ports
• expensive	• cost effective
• versatility	• for applications in which cost, power and space are critical
• general-purpose	• single-purpose

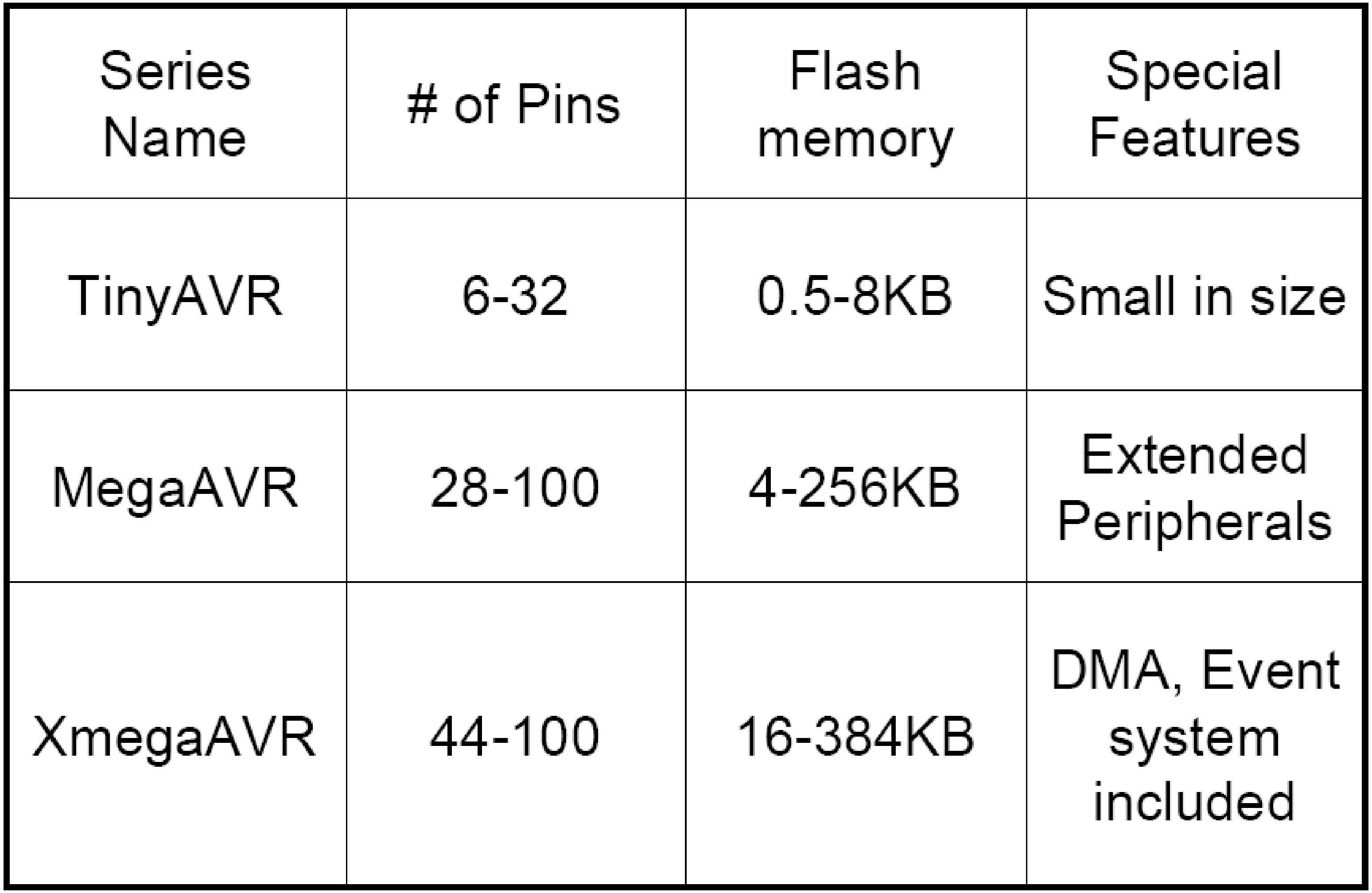
List of Microcontrollers

- **1972** - Texas Instrument TMS 1000, 1st single µC, 4-bit
- **1976** - Intel 8048, 8-bit µC, 1k ROM, 64b RAM, 27 I/O
- **1980** - Intel 8051, 4k ROM, 128b RAM, 32 I/O, 2 16-bits timers
- **1980s**
- (MCS-51 family)
  - Intel 8031, 8052, 8751, …
  - Atmel AT89C51, AT 89C1052/2051, **Atmel AVR**
  - Dallas Semiconductor DS5000 series
  - Philips, National Semiconductor, …
  - (Other µCs) Microchip PIC16 series, Motorola 68HC11, Zilog's Z86

AVR Microcontrollers

- AVR (**A**lf Egil Bogen and **V**egard Wollan's RISC processor) or AVR (**A**dvanced **V**irtual **R**ISC) microcontrollers are available in three categories:
- **Tiny AVR –**
  - Less memory, small size, suitable only for simpler applications.
- **Mega AVR –**
  - These are the most popular ones having good amount of memory (up to 256 KB), higher number of inbuilt peripherals and suitable for moderate to complex applications.
- **Xmega AVR –**
  - Used commercially for complex applications, which require large program memory and high speed.

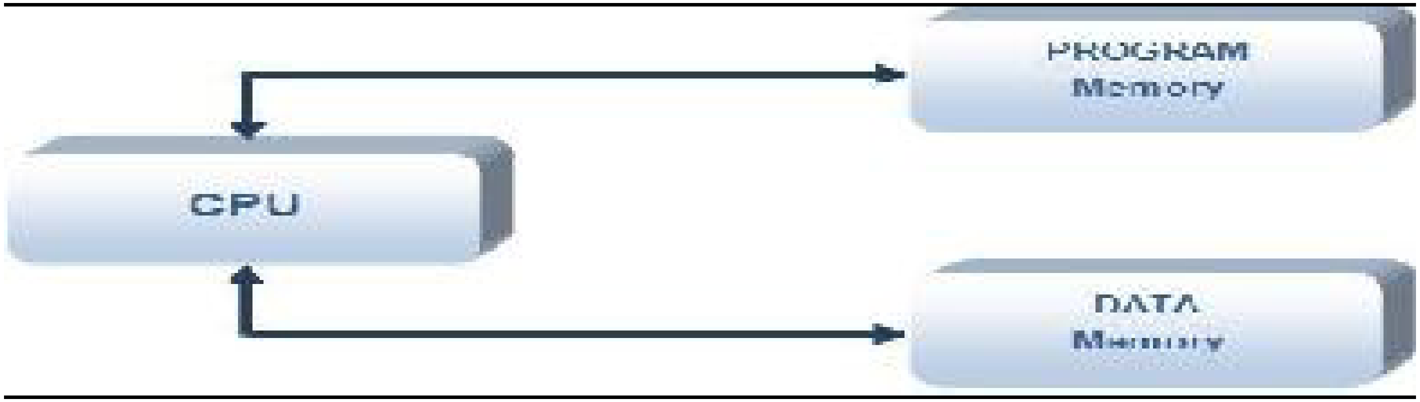
AVR Microcontrollers



Series Name	# of Pins	Flash memory	Special Features
TinyAVR	6-32	0.5-8KB	Small in size
MegaAVR	28-100	4-256KB	Extended Peripherals
XmegaAVR	44-100	16-384KB	DMA, Event system included

AVR Microcontrollers Features

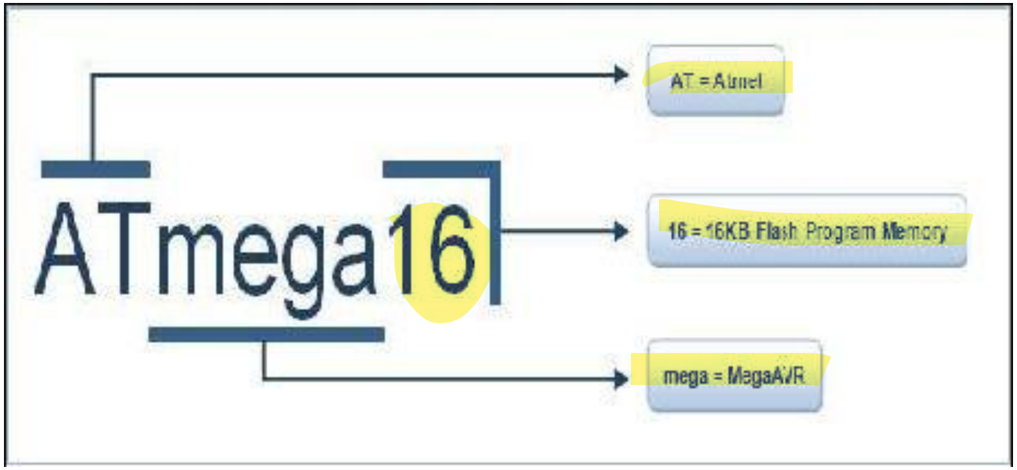
- The **AVR --- a 8-bit RISC single chip** microcontroller developed by Atmel in 1996 --- uses modified Harvard architecture .
- In Harvard architecture, **program** and **data** are stored in separate physical memory systems that appear in different address spaces.



AVR Microcontrollers Features

- The AVR was one of the first microcontroller families to use on-chip flash memory for program storage.

- Whereas one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

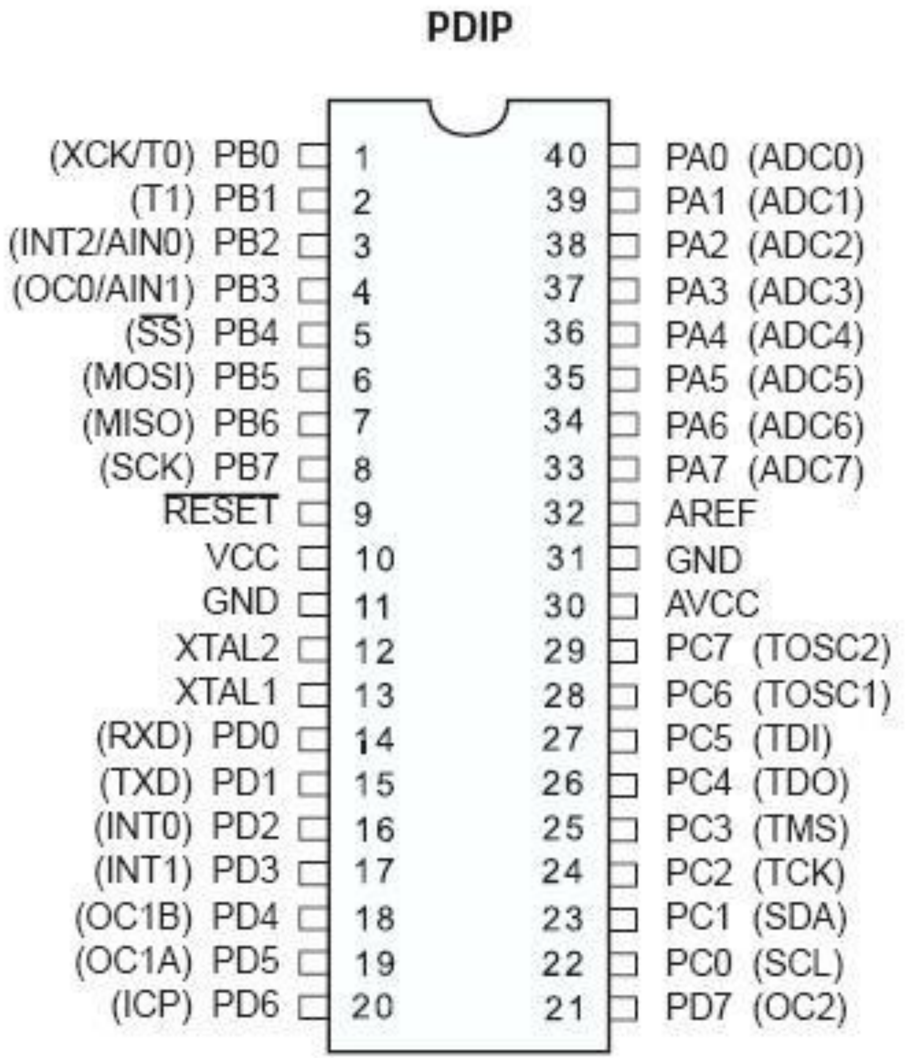


AVR Microcontroller Naming Convention

ATMega16: Features

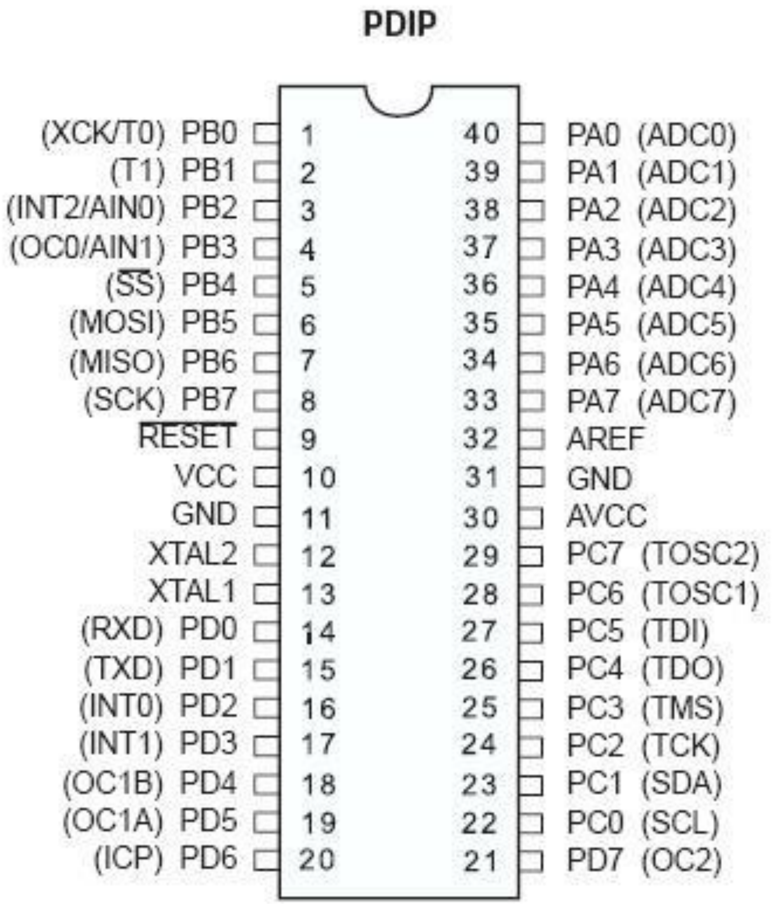
- 16 Kbytes of in-System Self-programmable Flash program memory
- 512 Bytes EEPROM
- 1 Kbyte Internal SRAM
- 32 8-bit General Purpose Working Registers
- 32 Programmable I/O Lines out of total 40 Pins DIP
- 8-channel, 10-bit ADC
- Two 8-bit Timer/Counters
- One 16-bit Timer/Counter
- Programmable Serial USART (Universal Synchronous Asynchronous Receiver Transmitter)
- 4 PWD (Pulse Width Modulator) Channel
- Operating Voltages: 4.5V - 5.5V
- Speed Grades: 0 - 16 MHz

ATMega16: Pin Descriptions



ATMega16: Pin Descriptions

- **GND:** Ground (0V). Note there are 2 ground Pins.
- **VCC:** Digital supply voltage. (+5V)
- **AVCC:** AVCC is the supply voltage pin for Port A and the A/D Converter.
- **AREF:** AREF is the analog reference pin for the A/D Converter.

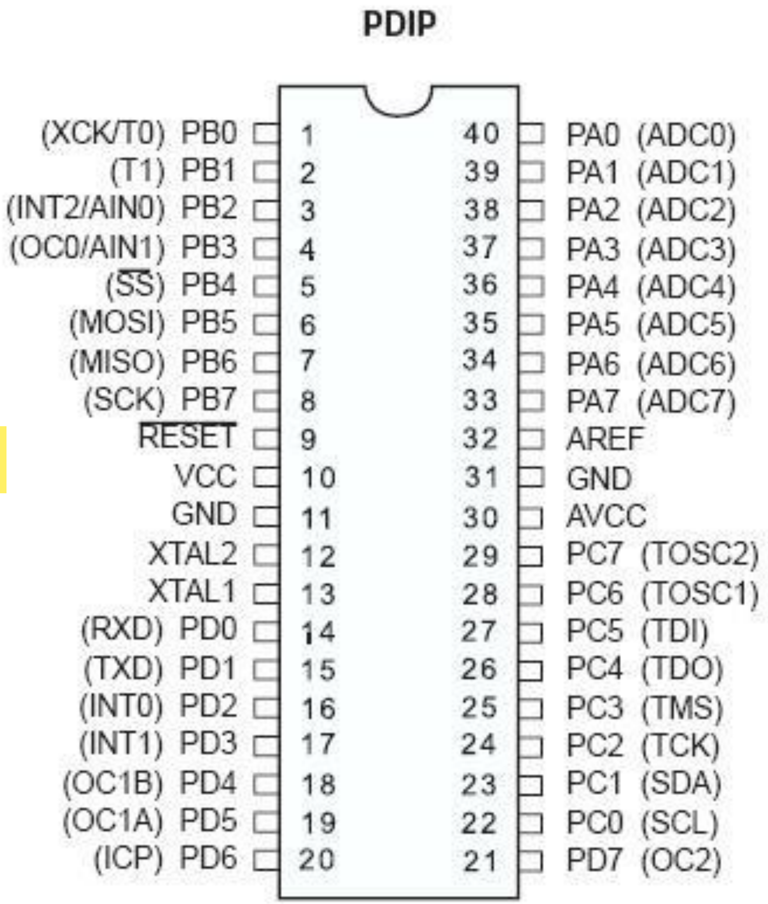


ATMega16: Pin Descriptions

- **XTAL1:** External oscillator pin 1

- **XTAL2:** External oscillator pin 2

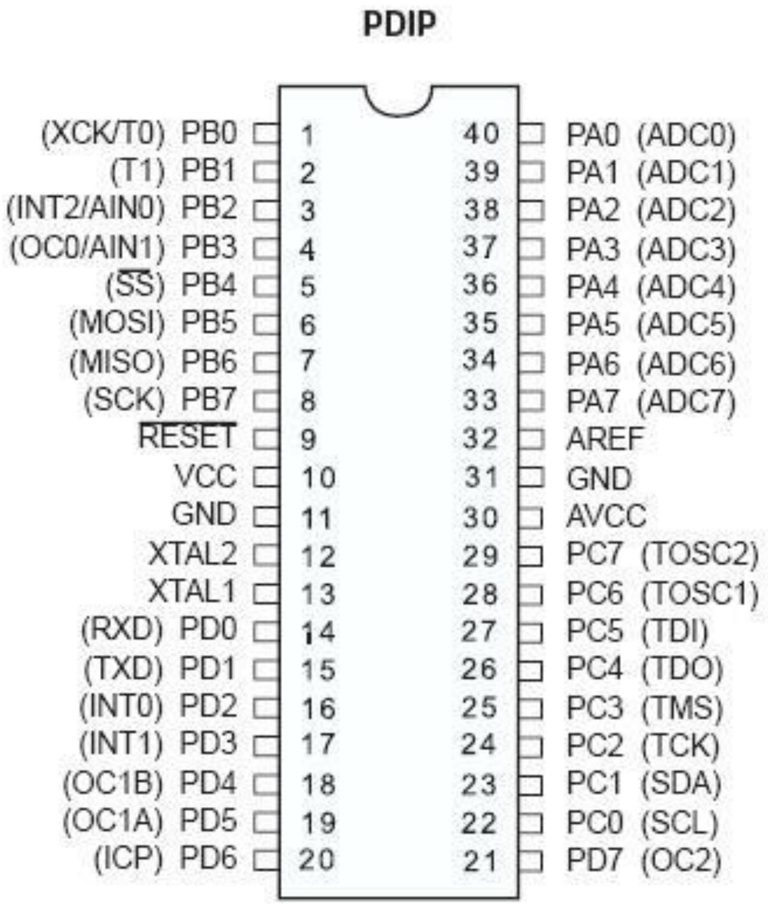
- **Port A (PA0 – PA7):** Port A serves as the analog inputs to the A/D Converter.
- It also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used.



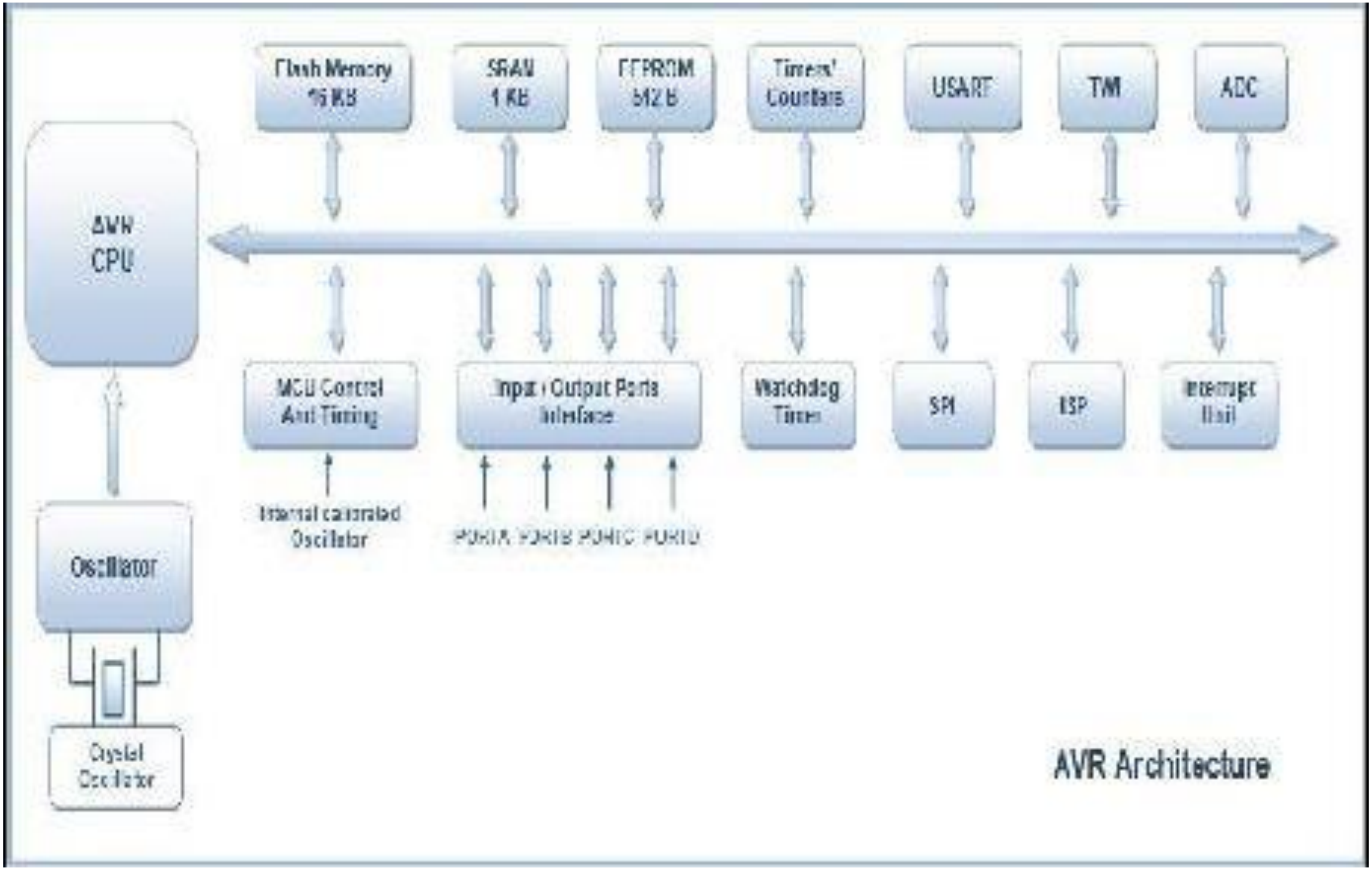
ATMega16: Pin Descriptions

- **Port B (PB0 – PB7):** Port B is an 8-bit bi-directional I/O port.
- **Port C (PB0 – PB7):** Port B is an 8-bit bi-directional I/O port.
- **Port D (PD0 – PD7):** Port B is an 8-bit bi-directional I/O port.

- Port B, C and D also serve the functions of various special features of the ATmega16.



ATMega16: Architectural Diagram



ATMega16: Architectural Diagram

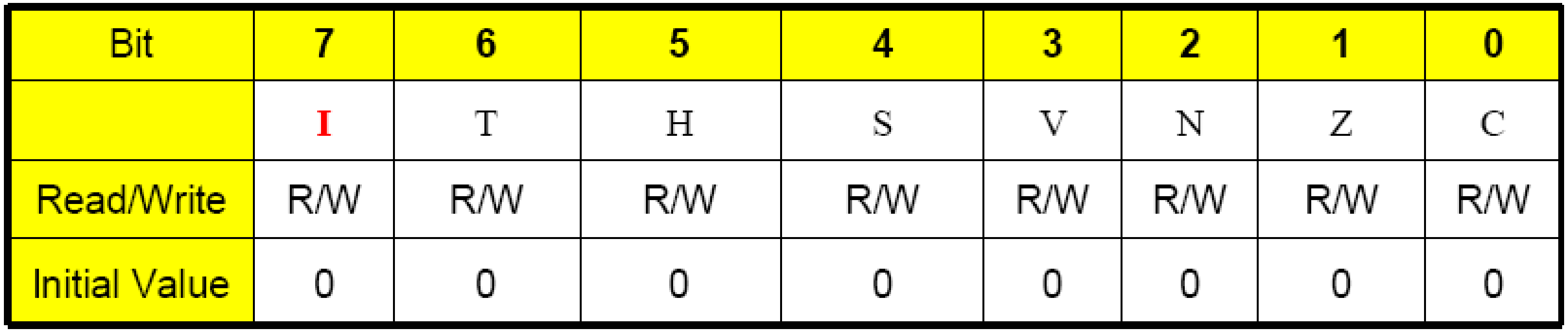
- **Oscillator:**
  - The ATMega16 can use an internal or external clock signal.
  - By default Atmega16 is set to operate at internal calibrated oscillator of 1 MHz.
  - The internal clock is an RC oscillator programmable to 1, 2, 4, or 8 MHz
  - The maximum frequency of internal oscillator is 8Mhz.
  - Alternatively, ATmega16 can be operated using an external crystal oscillator with a maximum frequency of 16MHz.

ATMega16: Architectural Diagram

- **CPU:**
  - PC: Program Counter
    - Address of next instruction
  - IR: Instruction Register
    - Pre-fetched instruction
  - ID: Instruction decoder
    - Current instruction
  - General Purpose Register (GPR): R0-R31
    - 8-bit x 32 GPR pipeline operation possible.
  - ALU (Arithmetic Logic Unit)
  - Status Register

ATMega16: Architectural Diagram

- **CPU – Status Register:**

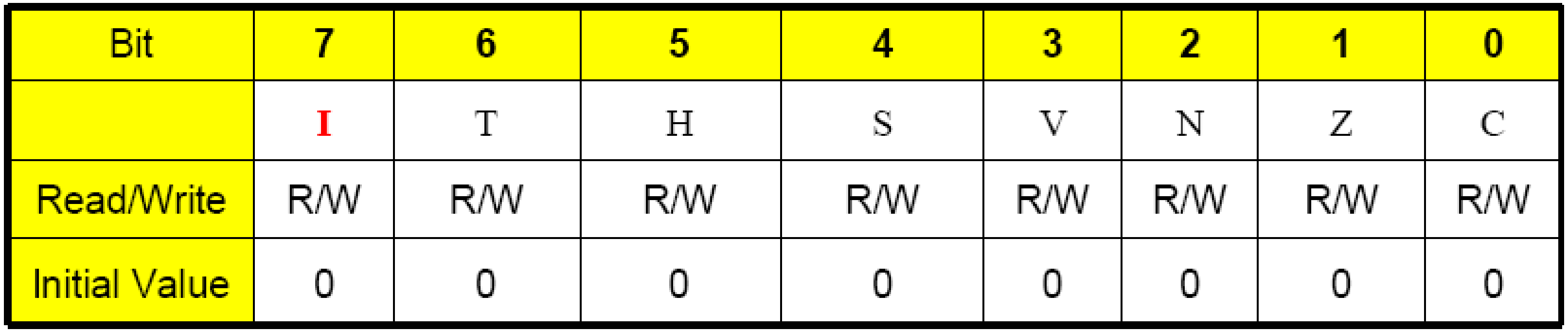


Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 0 - C: Carry Flag**
  - The Carry Flag C indicates a carry in an arithmetic or logic operation.
- **Bit 1 - Z: Zero Flag**
  - The Zero Flag Z indicates a zero result in an arithmetic or logic operation.

ATMega16: Architectural Diagram

- **CPU – Status Register:**

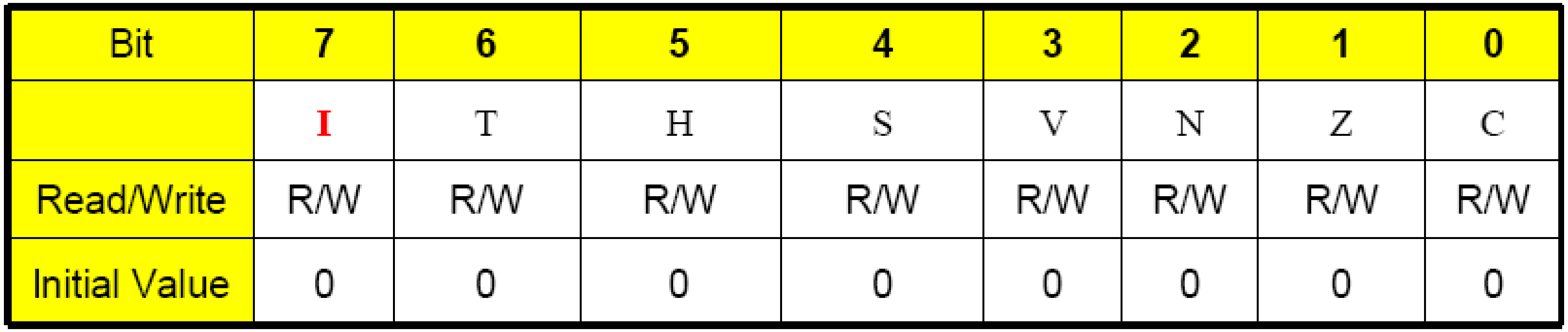


Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 2 - N: Negative Flag**
  - The Negative Flag N indicates a negative result in an arithmetic or logic operation.
- **Bit 3 - V: Two’s Complement Overflow Flag**
  - The Two's Complement Overflow Flag V supports two's complement arithmetic

ATMega16: Architectural Diagram

- **CPU – Status Register:**

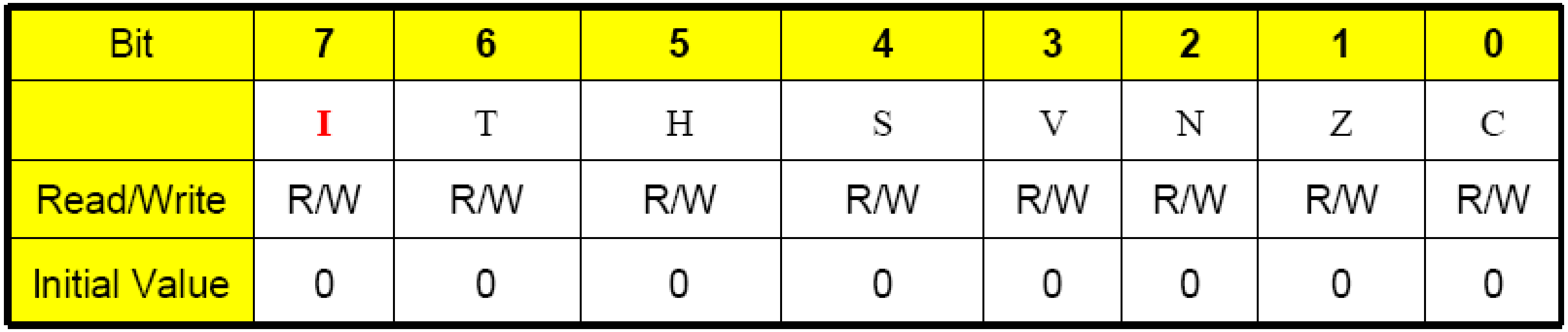


Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 4 - S: Sign Bit Flag (S = N ⊕ V)**
  - The S bit is always an XOR between the N and V value.
- **Bit 5 - H: Half-carry Flag**
  - The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry is useful in BCD arithmetic.

ATMega16: Architectural Diagram

- **CPU – Status Register:**



Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 6 - T: Bit Copy Storage Flag**
  - The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit.
- **Bit 7 - I: Global Interrupt Enable Flag**
  - The Global Interrupt Enable bit must be set for the interrupts to be enabled.

ATMega16: Architectural Diagram

- **Timers/Counters:**
  - Most commonly used complex peripheral
  - Atmega16 consists of **two** 8-bit and **one** 16-bit timer/counter.
  - Think of them as binary up-counters
    - In timing mode, count time periods
    - In counting mode, counting events or pulses
  - Timers are useful for generating precision actions e.g., for creating time delays between two operations.

ATMega16: Architectural Diagram

- **USART:**
  - **Universal Synchronous Asynchronous Receiver and Transmitter** interface is available for interfacing with external device capable of communicating serially (data transmission bit by bit).
- **Two Wire Interface (TWI):**
  - TWI can be used to set up a network of devices
    - Many devices can be connected over TWI interface forming a network
    - The devices can simultaneously transmit and receive and have their own unique address.

ATMega16: Architectural Diagram

- **ADC Interface:**
  - Atmega16 is equipped with an 8 channel ADC (**Analog to Digital Converter) with a** *resolution* of 10-bits.
  - ADC reads the analog input for e.g., a sensor input and converts it into digital information which is understandable by the microcontroller.
- **Serial Peripheral Interface (SPI):**
  - SPI port is used for serial communication between two devices on a common clock source.
  - The data transmission rate of SPI is more than that of USART.

Why Was Arduino Developed?

- Physical Computing – using components that can interact with people and with the world around us
- The Arduino was <u>originally</u> developed for artists and designers to prototype interactive displays
  - Developed for non-scientists
  - Minimalist programming
  - "Forgiving" circuitry that can handle a wide variety of wiring errors

Different Varieties of Arduino

ARDUINO UNO

ARDUINO 101

ARDUINO PRO

ARDUINO MICRO

ARDUINO PRO MINI

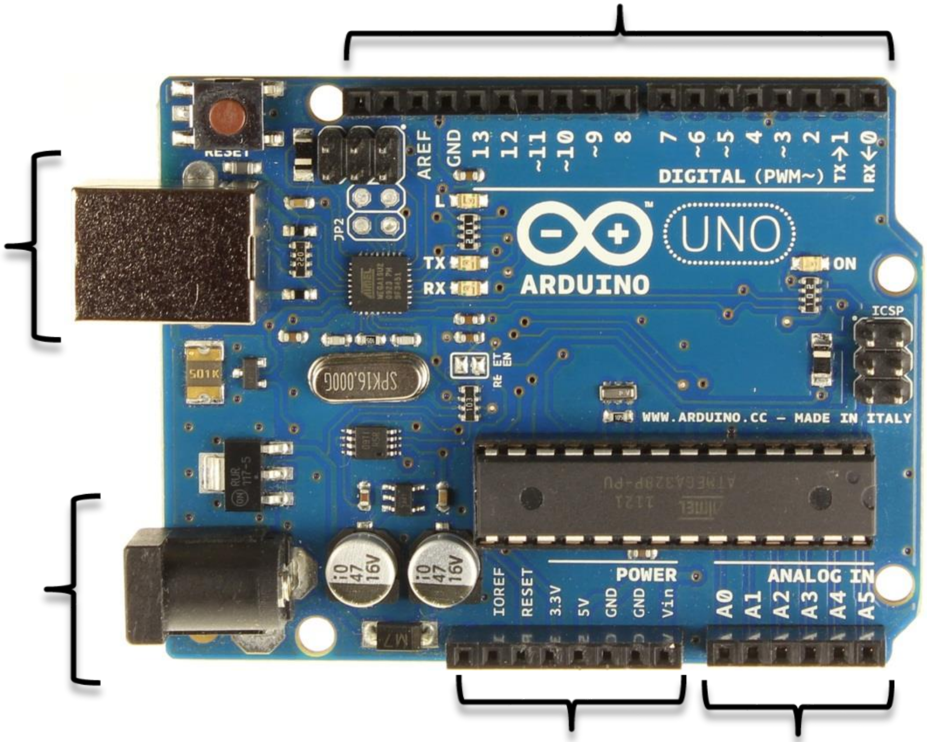
ARDUINO NANO

ARDUINO MEGA 2560

ARDUINO ZERO

ARDUINO DUE

Digital Input / Digital output (PWM on pins 3, 5, 6, 9, 10, 11)



Arduino Uno Revision 3

7-12 volt input power (9v is common)

Power pins

Analog input / Digital input or output

What Can Arduino Uno Be Used To Teach?

- Introductory electronics (voltage, current, resistance)
- How sensors and actuators work
- Simple programming
- Design of basic scientific equipment
- Troubleshooting
- Challenges of communicating with users through a project (e.g., messages, formatting numbers, ease of use, etc.)
- Statistics and variation in data gathering

What Can Students Do?

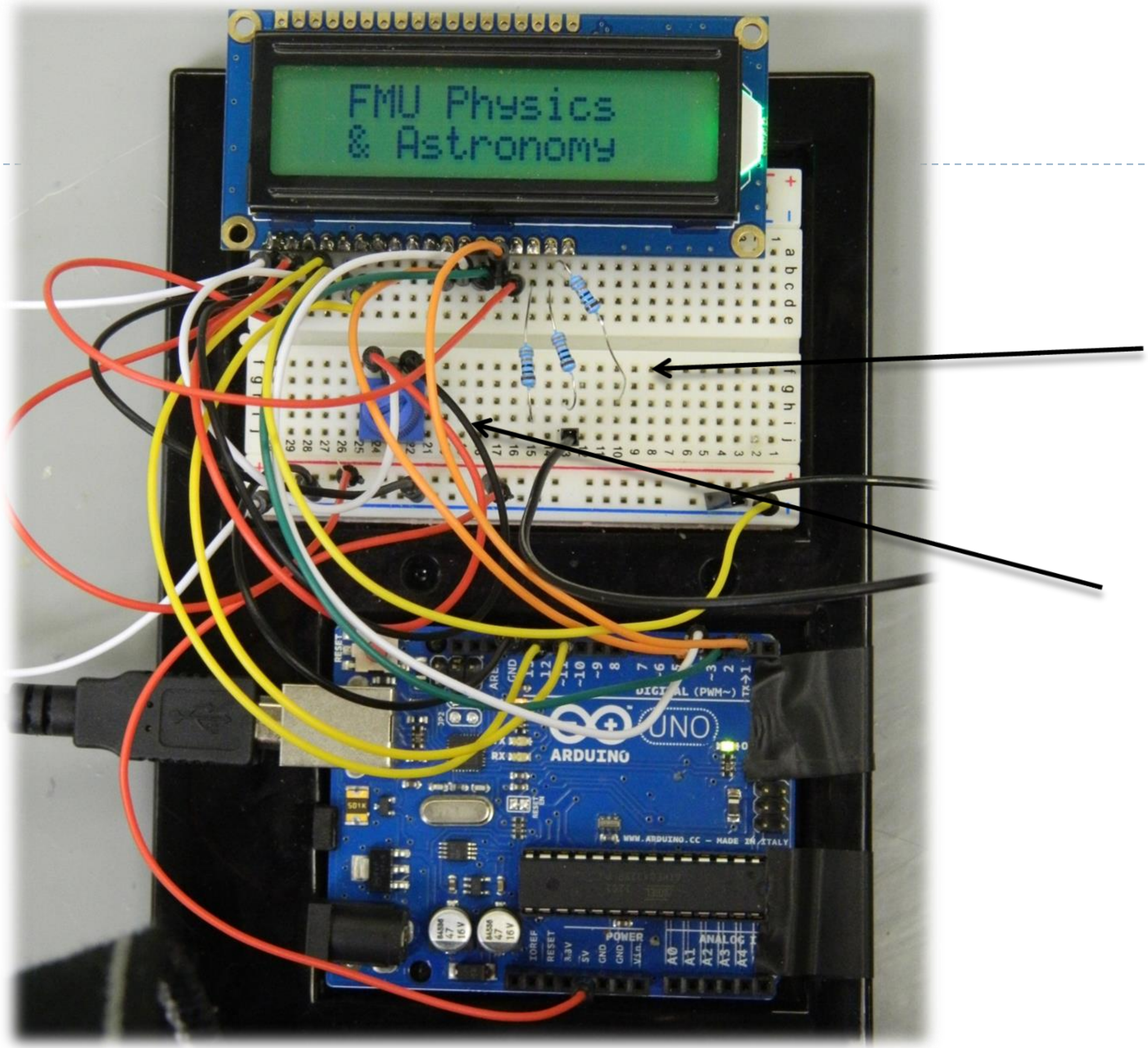
- Quickly able to prototype a working project
- Able to produce a working computer program
- The tactile "feel" of assembling a project is very rewarding; making something that works
- FINALLY understanding that a scientific instrument is only as good as its design and calibration
- Get to troubleshoot circuits under the tutelage of the instructor to develop a logical, orderly method.

What Are Some Good Components to Start With?

- LEDs….lots of LEDs with variety of colors (be sure to also acquire lots of current-limiting resistors)
- Good for blinking, dimming using PWM, “Knight Rider” effect with 8 or more LEDs, strobe effects, etc.

What Are Some Good Components to Start With?

- LCD Screen with backlight (16x2 is common and teaches the complications of textual display)
- The parallel version uses 4 digital pins for display
- Instructions for how to connect is built in as a sample “sketch” in the Arduino
- Students get a <u>huge</u> degree of satisfaction from transmitting text to the LCD screen

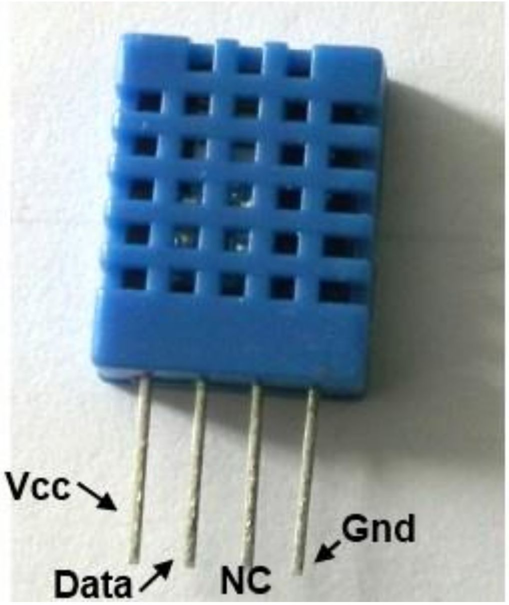


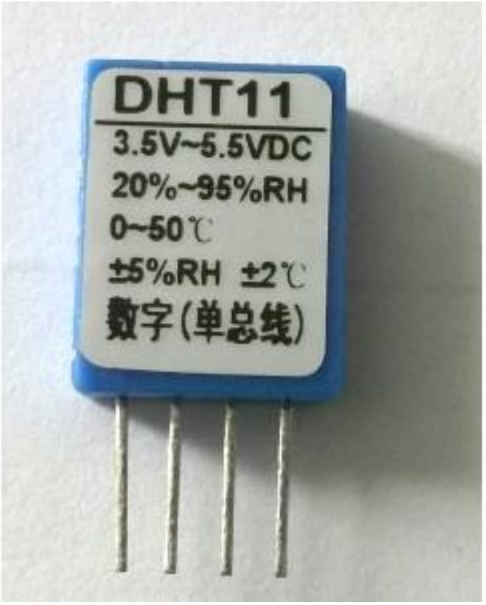
Current-limiting resistors for RGB backlight

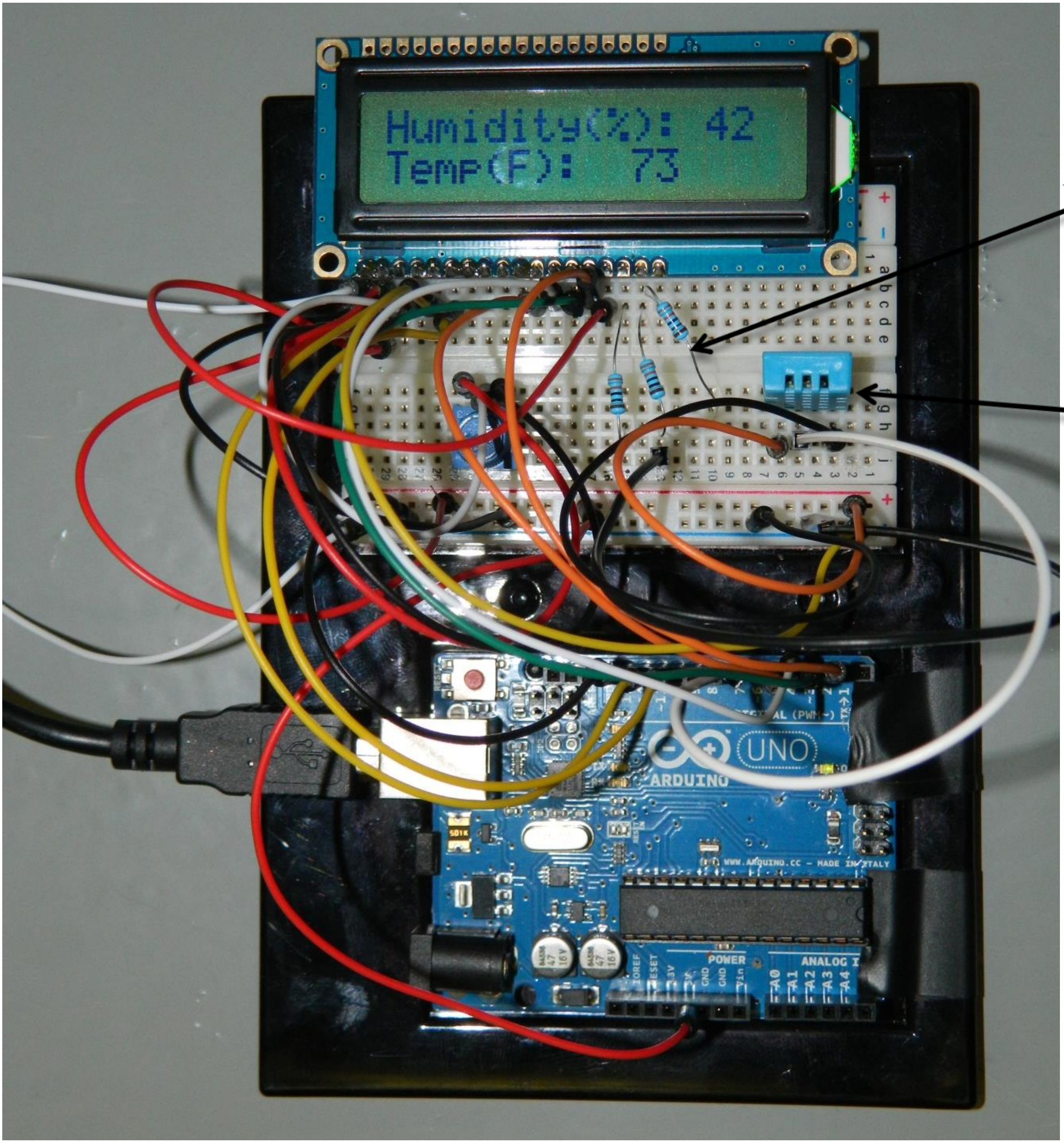
10 kΩ potentiometer to adjust screen contrast

What Are Some Good Components to Start With?

- Basic temperature and/or humidity sensor
- DHT11 or DHT22 is a good entry-level choice
- Library can be downloaded and incorporated to give easy access to features of the sensor





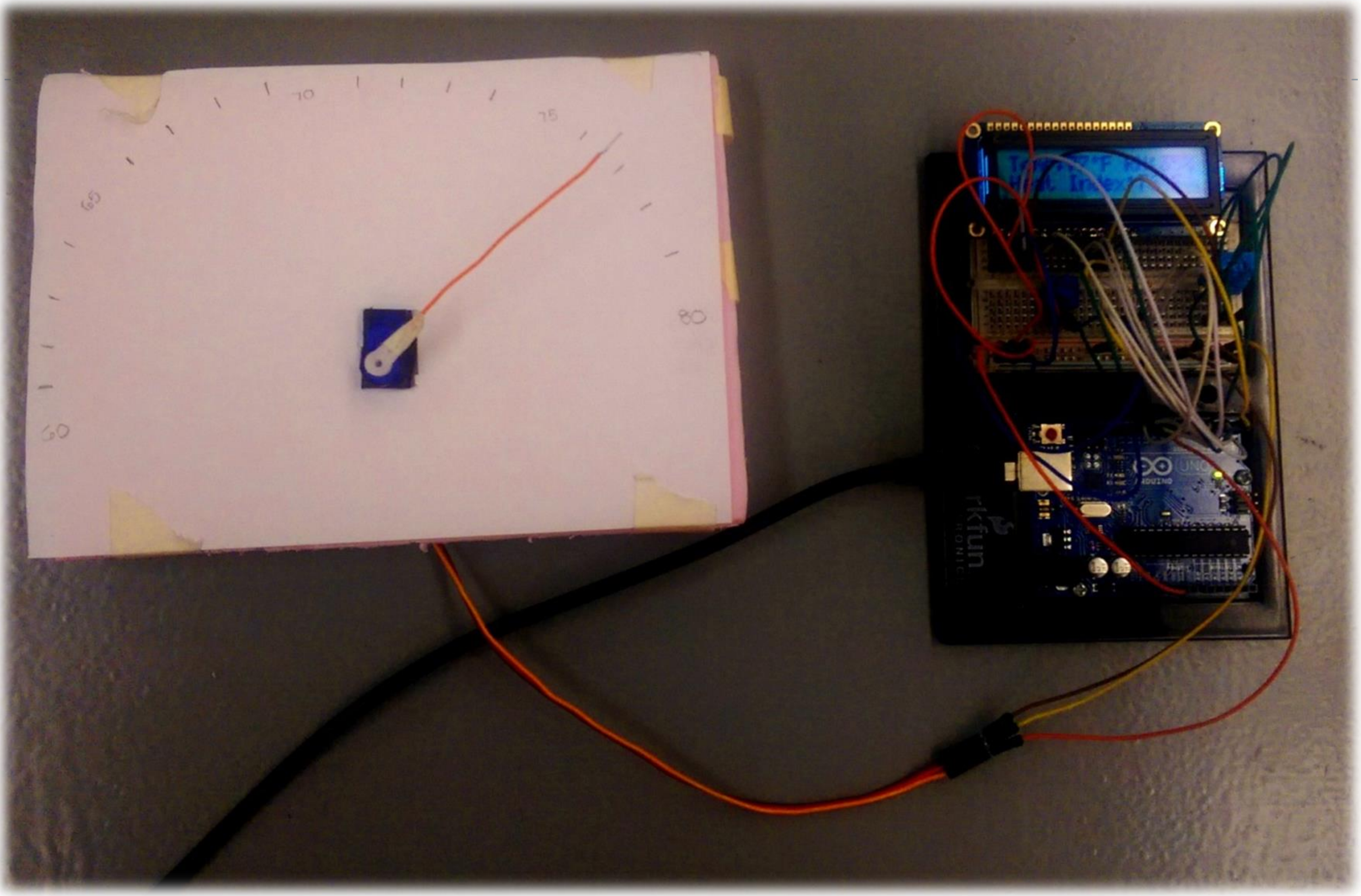


Current-limiting resistors for RGB backlight

DHT 11 Sensor

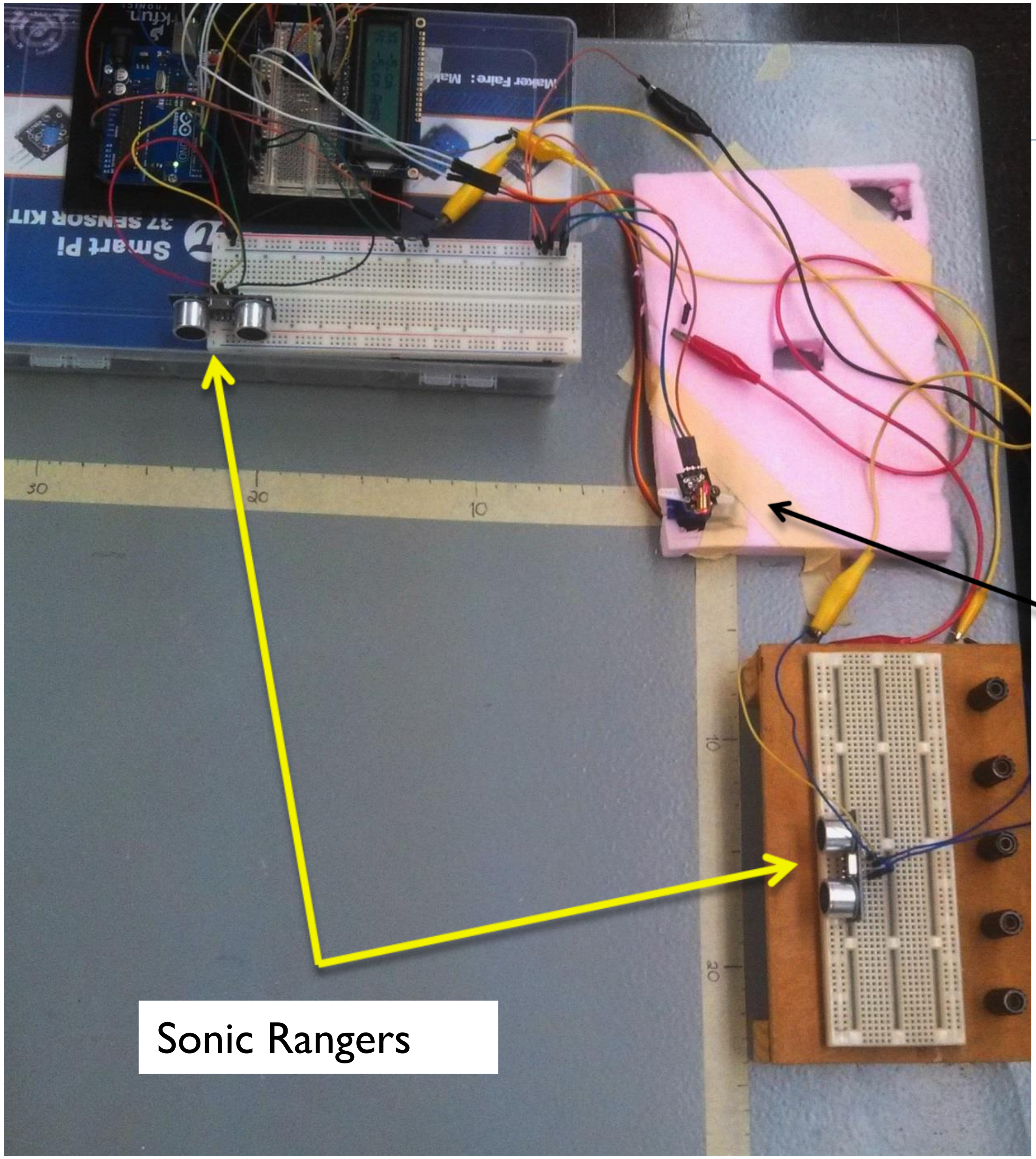
What Are Some Good Components to Start With?

- Small servos
- Arduino can be used to position servo at a given angle for use in projects.
- Other sensors can be attached to the servo to add a layer of complexity
- Servo library included in Arduino IDE


60
65
70
75
80
UNO
rkfun

What Are Some Good Components to Start With?

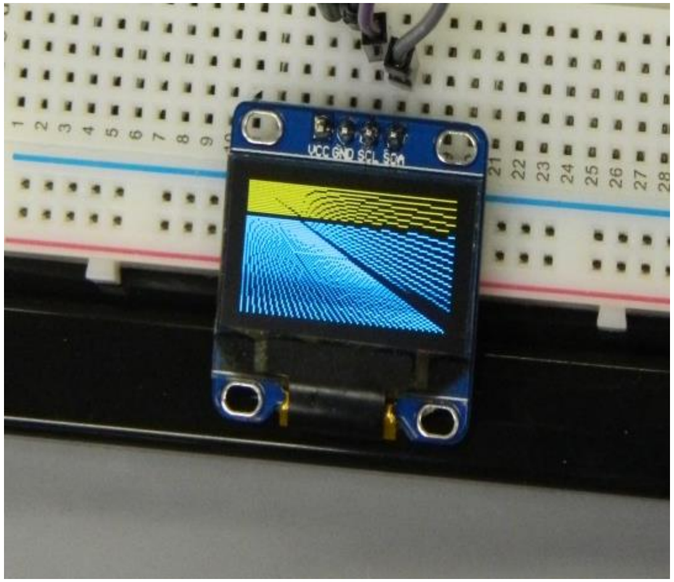
- Ultrasonic Ranger – uses ultrasonic waves to determine the distance between sensor and object based on time between emission of wave and echo of it.
- Good for introducing discussion of speed of sound variation with environmental conditions
- Calibration principles are easily discussed

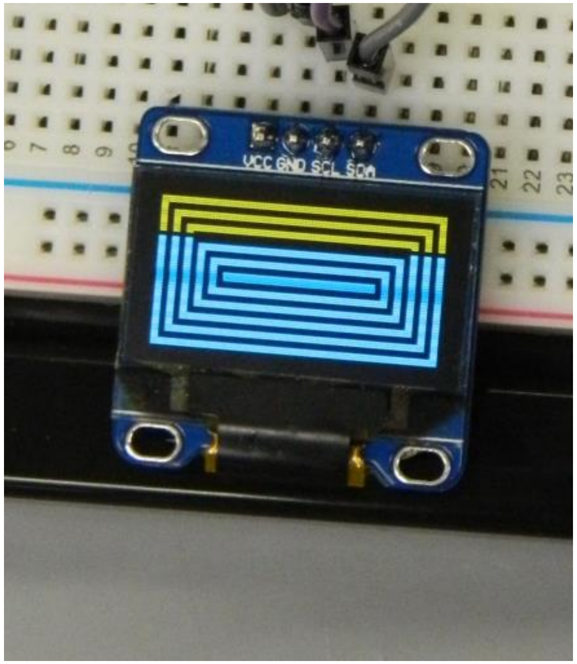


Laser pointer mounted on servo that aims at a target based on readings of dual sonic rangers

What Are Some Other Components That Are Relatively Easy to Use?

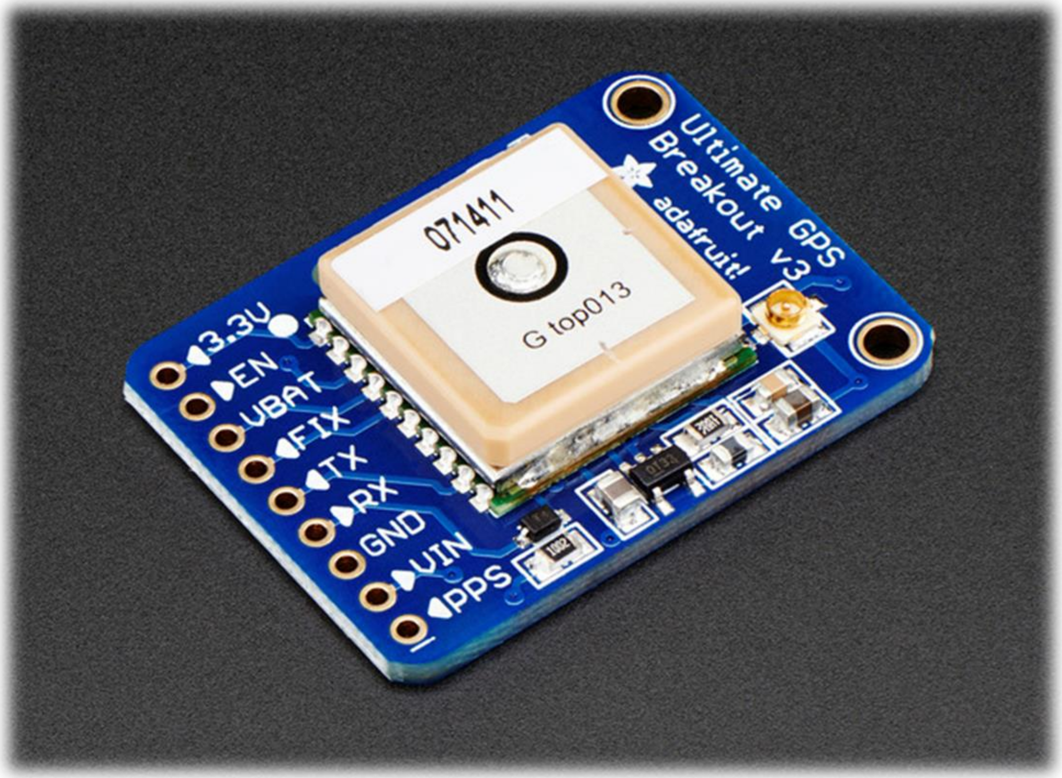
- 128x64 Organic LED screen that can handle graphics as well as text





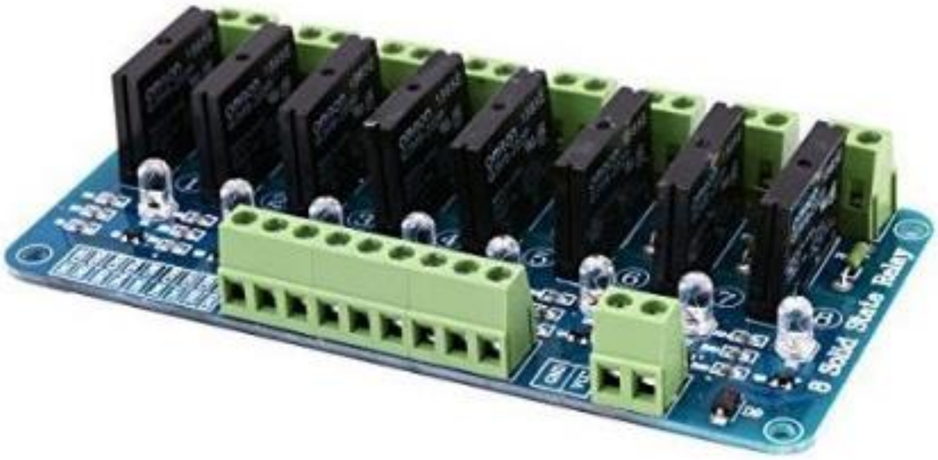
What Are Some Other Components That Are Relatively Easy to Use?

- GPS Module that can be used for a variety of purposes such as triggering a response only when the user is standing in a particular location

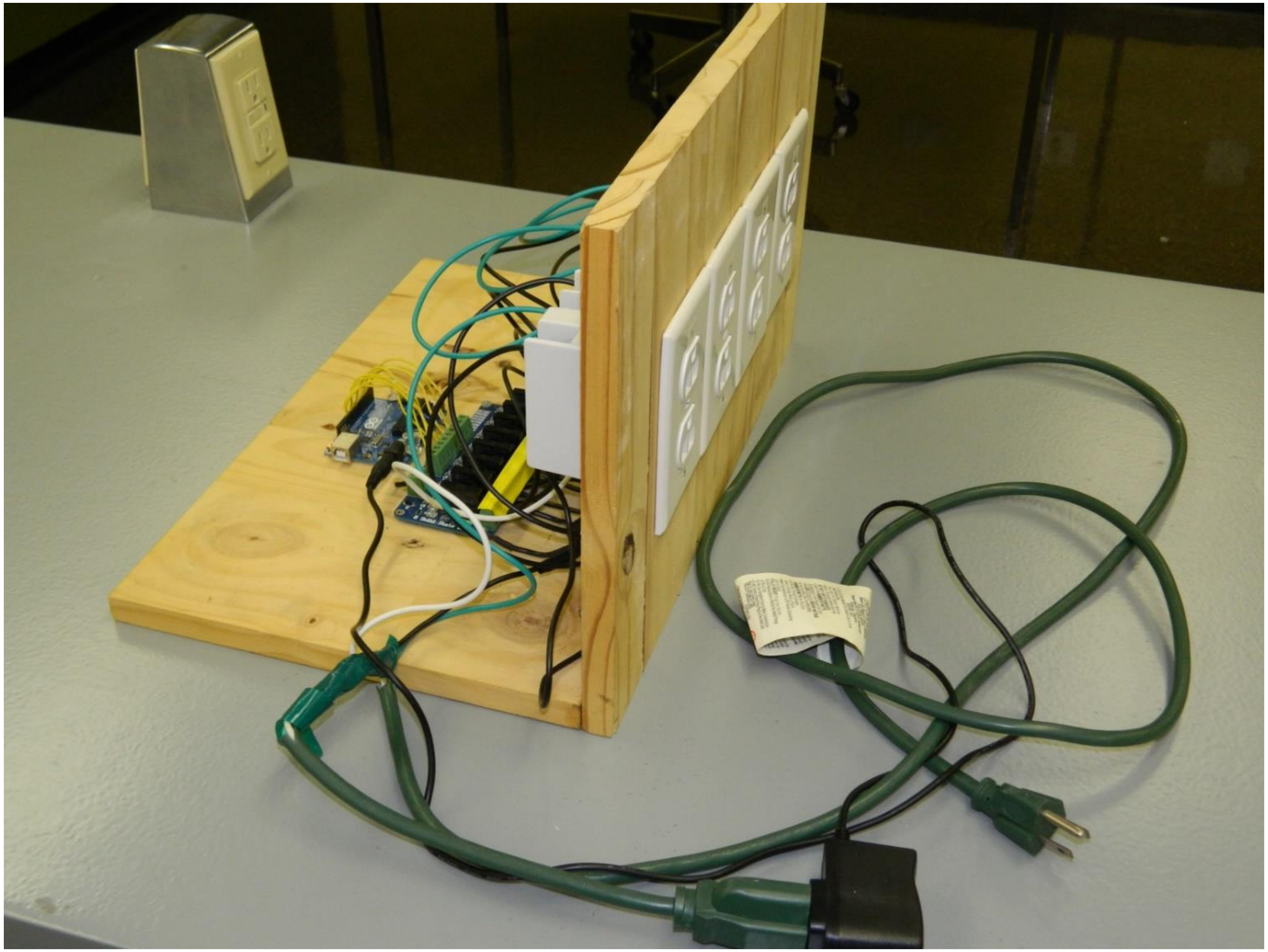


What Are Some Other Components That Are Relatively Easy to Use?

- Mechanical or Solid-State Relay Boards to control alternating current loads of several amps



AC Outlets Controlled by Arduino



Thank You !!