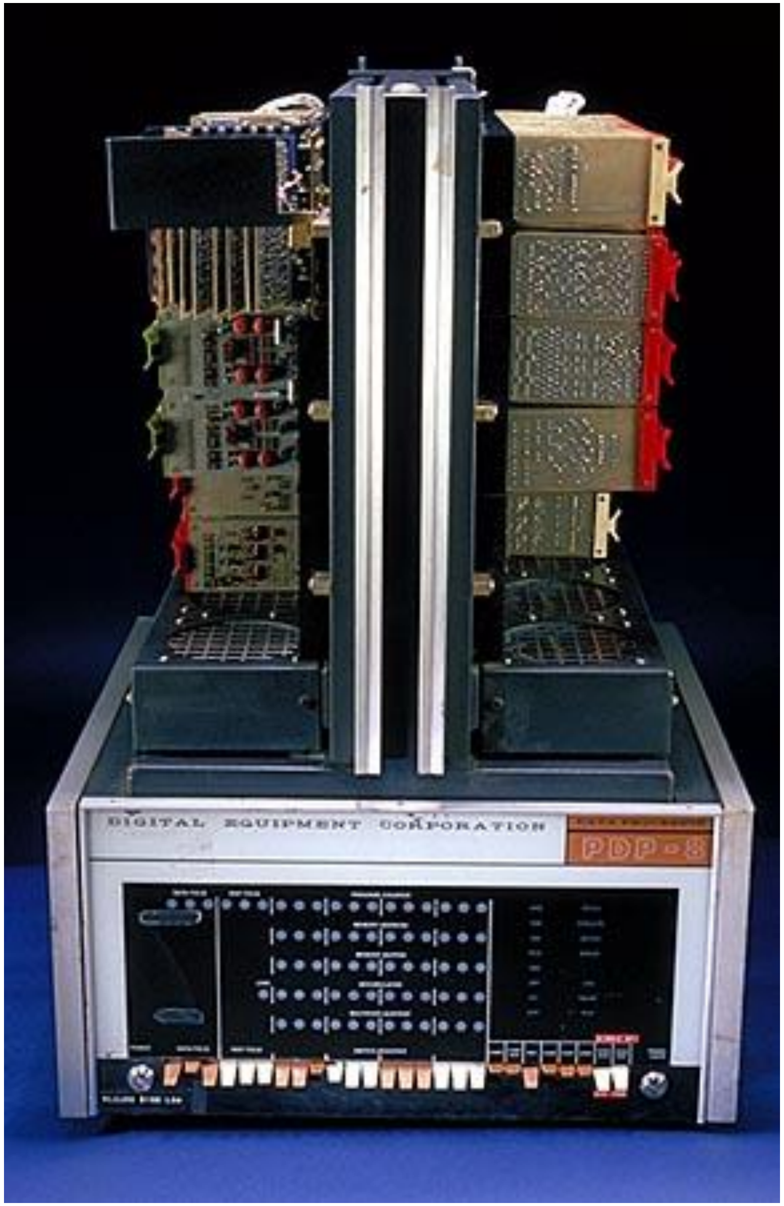
Mechatronics

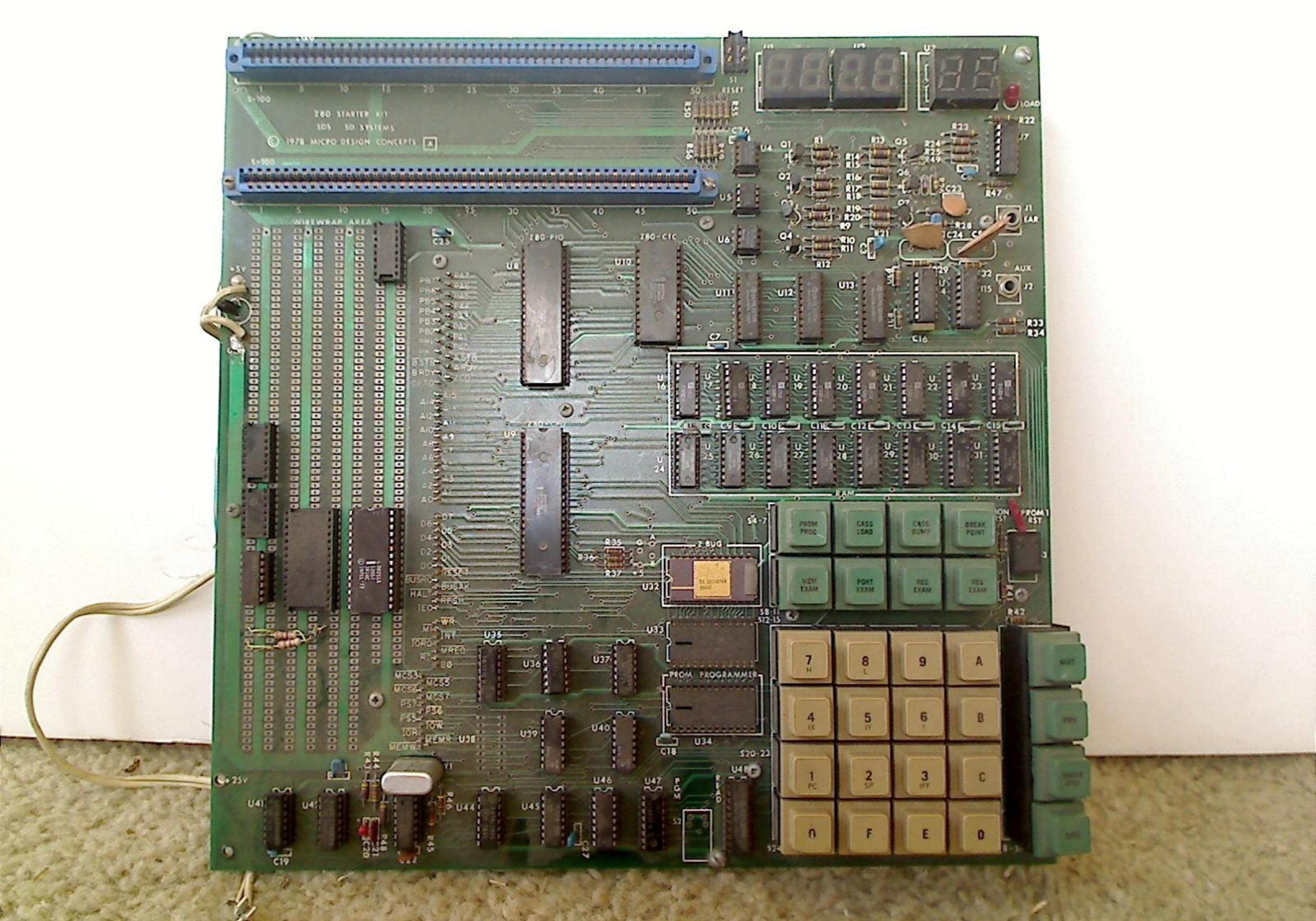
Using microcontrollers

Mechatronics is the combination of mechanical engineering, electronic engineering, computer engineering, software engineering, control engineering, and systems design engineering in order to design, and manufacture useful products. Mechatronics is a multidisciplinary field of engineering, that is to say it rejects splitting engineering into separate disciplines. Originally, mechatronics just included the combination between mechanics and electronics, hence the word is only a portmanteau of **mecha**nics and elec**tronics**. However, as technical systems have become more and more complex the word has been "updated" during recent years to include more technical areas.

IBM 650
My First Personal Computer

PDP-8





Example of a SBC or Single Board Computer

Microcontrollers

Microcontrollers are computers that are:

- Usually stand alone
- Have all basic computer functions; input, output, storage and decision ability
- Controlling program developed externally
- Usually have the Harvard Architecture where program and data storage areas are separate

Common types of current microcontrollers

Intel 8035 Considered the 1st introduced in 1976

Parallax Basic Stamp

Microchip PIC

Parallax Propeller

TI 430

AVR ATmega

ARM

Many others

Three functions required for Microcontrollers

1. Program Development- The program that will run on the microcontroller is developed using programs running on a PC. Once a program is developed, it is compiled into a machine lever language.

2. Download & Burn the program- The program developed in step 1 has to be downloaded and written or ‘burned’ into the microcontroller. This is called Programming the microcontroller.

3. Test the prototype- Once the program has been programmed into microcontroller, it has to be tested in the circuit for which it is intended. This can be either the actual circuit or in a prototyping board.

Integrated Development Environments IDE

- Software to create microcontroller programs
  - Usually can support multiple languages from various sources
  - Some can help ‘simulate’ the chip operation
- Software to download programs to the chip
- Method to download the program to the microcontroller

Signaling Project

Purpose: To implement an intelligent railroad signaling system to be installed at HALS.

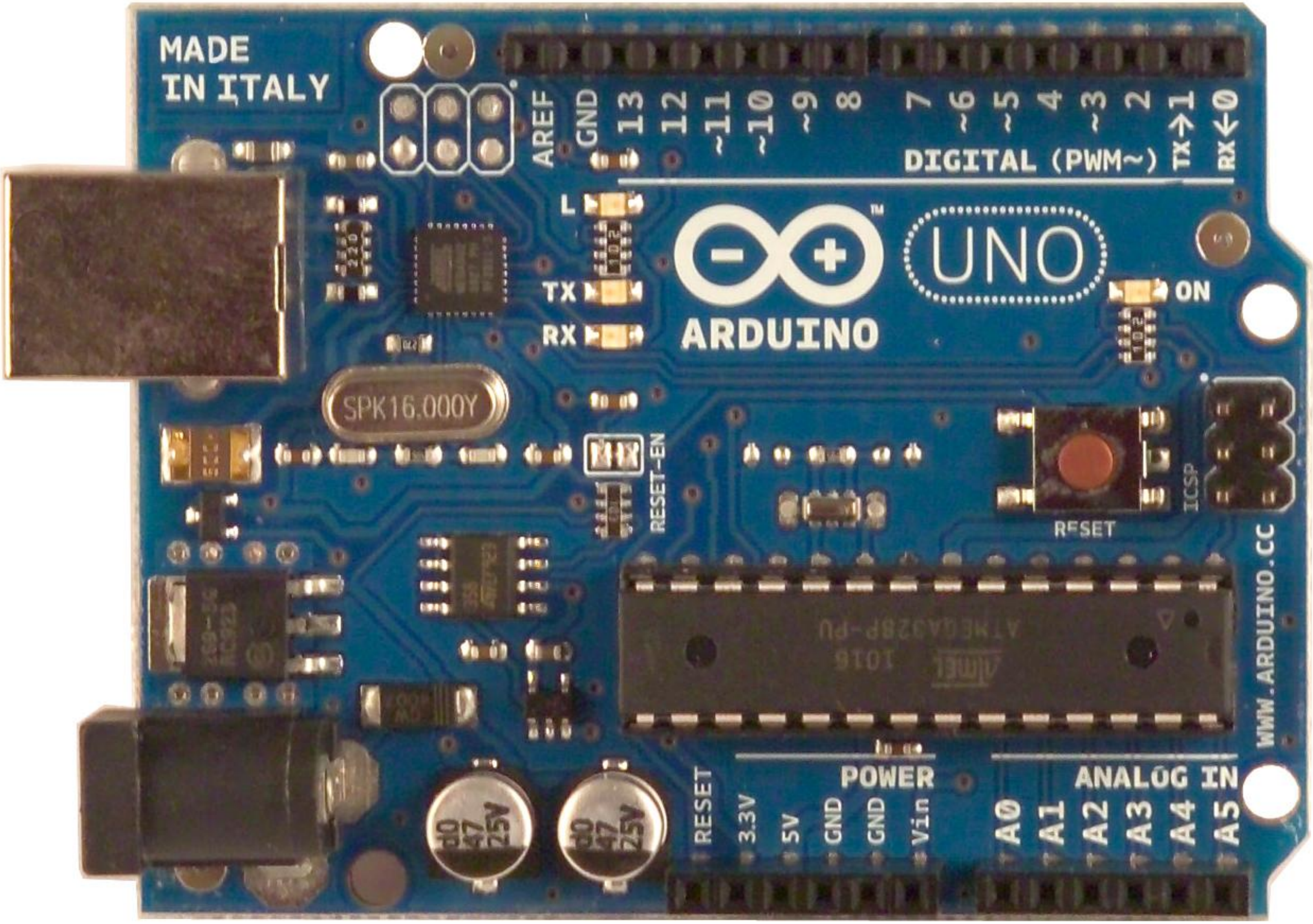
Selection Criteria for prototyping environment:

Analog input
serial I/O built in
quick program development/change
easily obtained
in circuit programming
built in voltage regs. from batteries
large advanced user base
Inexpensive development software

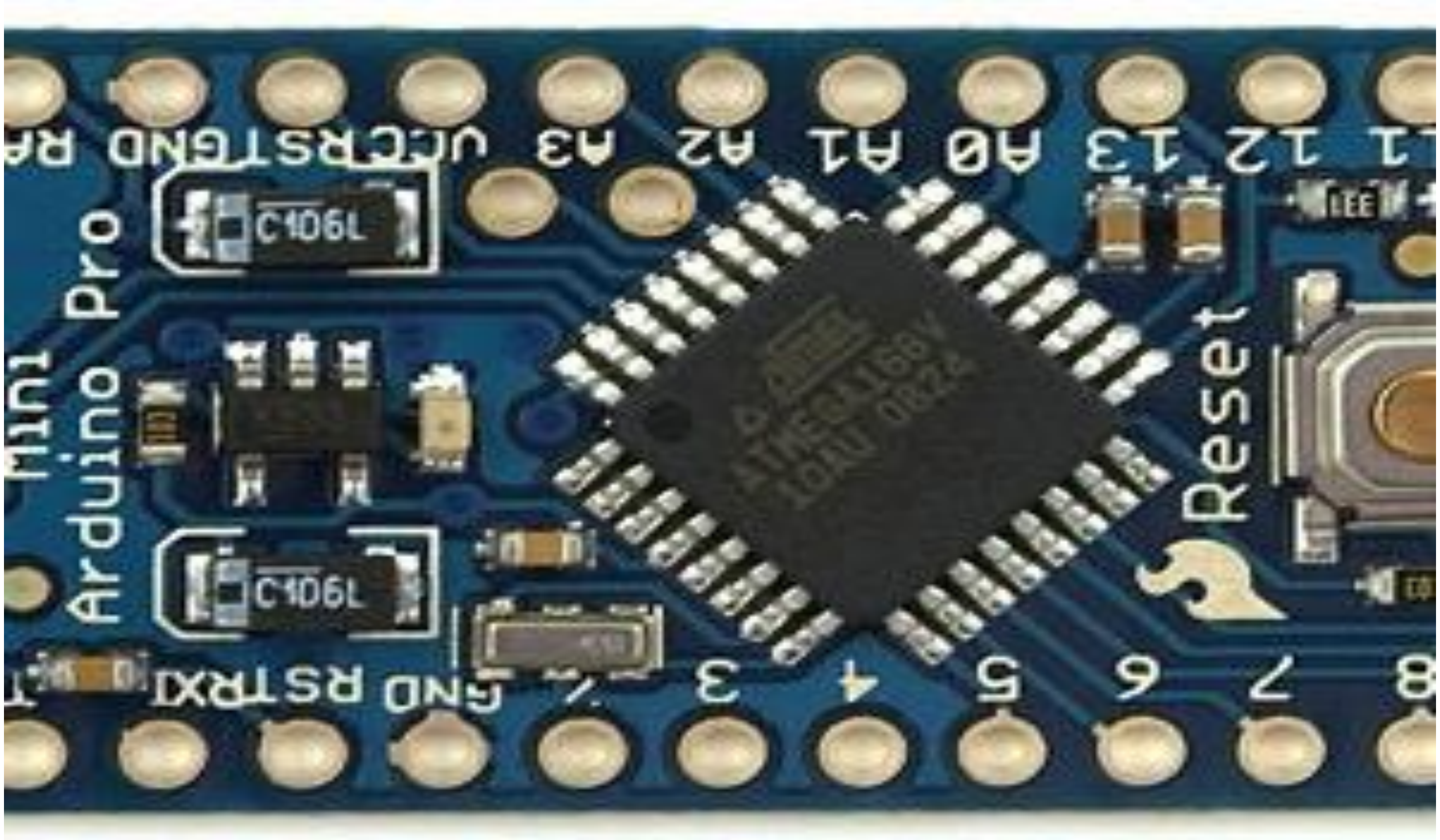
Arduino Environment around the AVR microcontroller

Reasons for choice:

- Totally integrated program, burn and development
- Large selection of compatible attachments
- Enhanced C language
- Wide range of usable examples
- Seamlessly move to lower level language
- Various sizes interchangeable



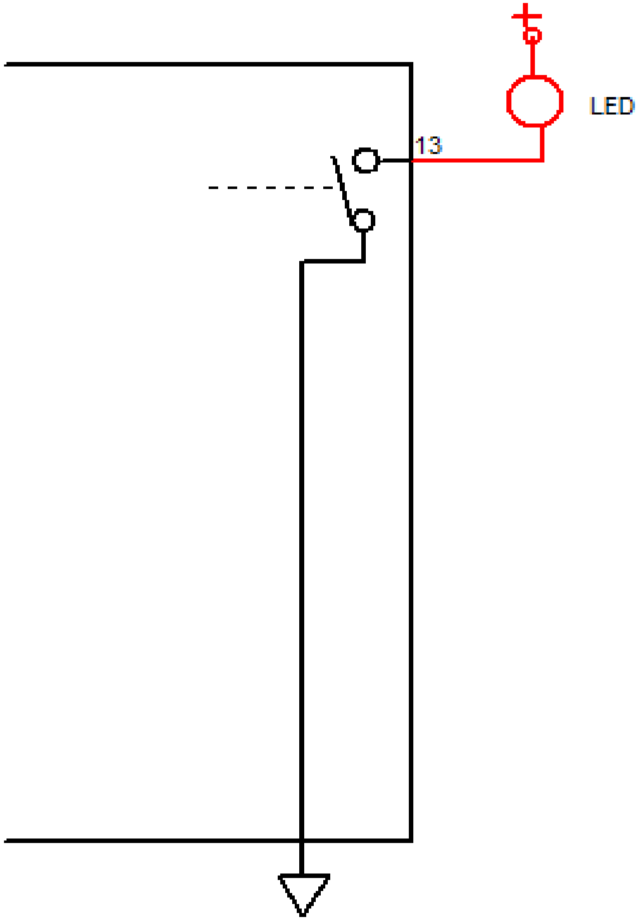
Standard Arduino



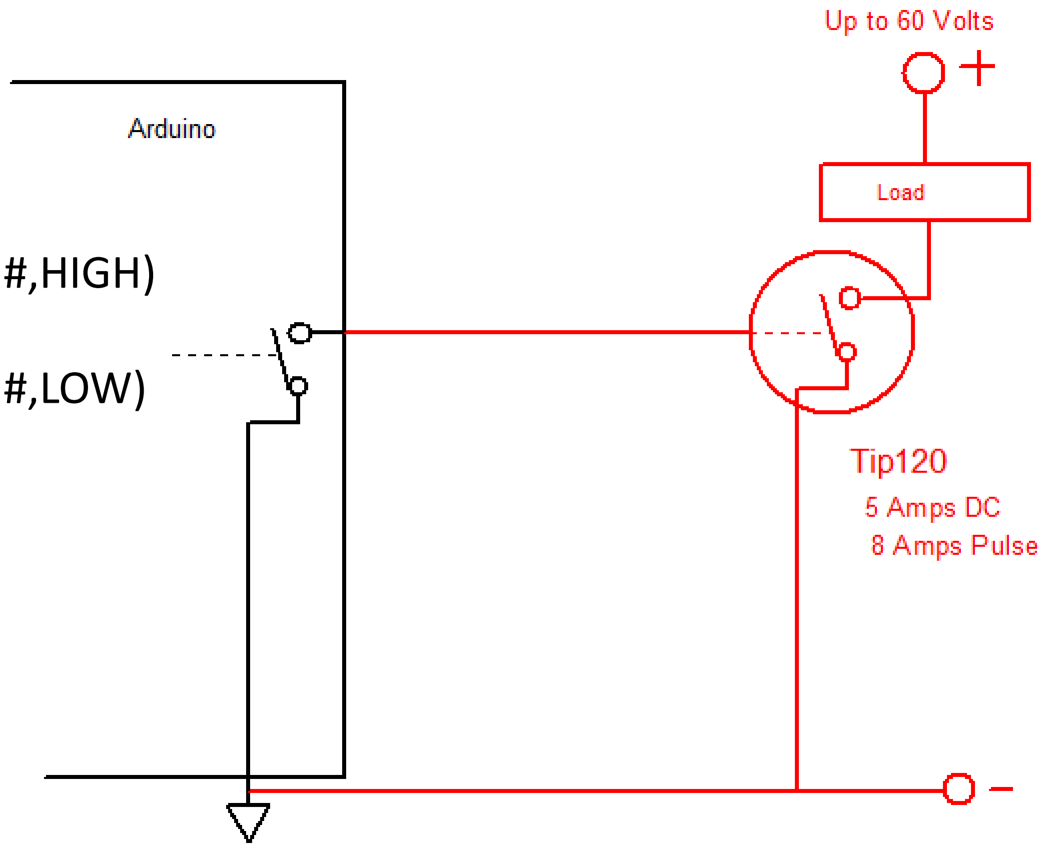
Arduino Pro Mini

Blink Sketch and Circuit

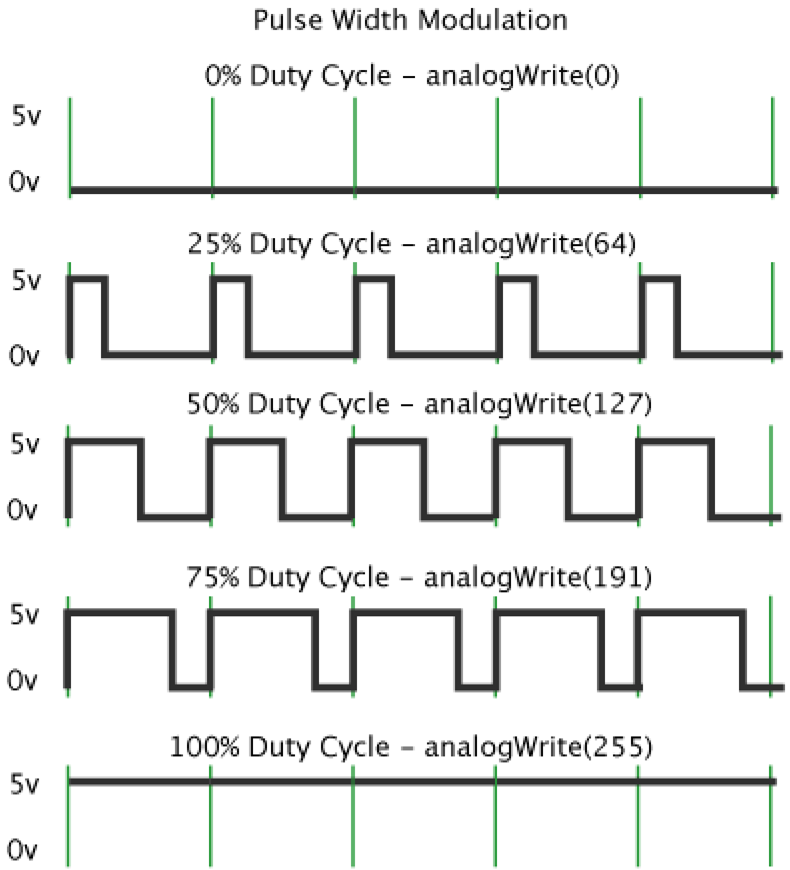
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```



DigitalWrite(pin#,HIGH)
Or
DigitalWrite(pin#,LOW)



PWM


Pulse Width Modulation
0% Duty Cycle - analogWrite(0)
5v
0v
25% Duty Cycle - analogWrite(64)
5v
0v
50% Duty Cycle - analogWrite(127)
5v
0v
75% Duty Cycle - analogWrite(191)
5v
0v
100% Duty Cycle - analogWrite(255)
5v
0v

Motor Control

0 % Speed

25% of full speed

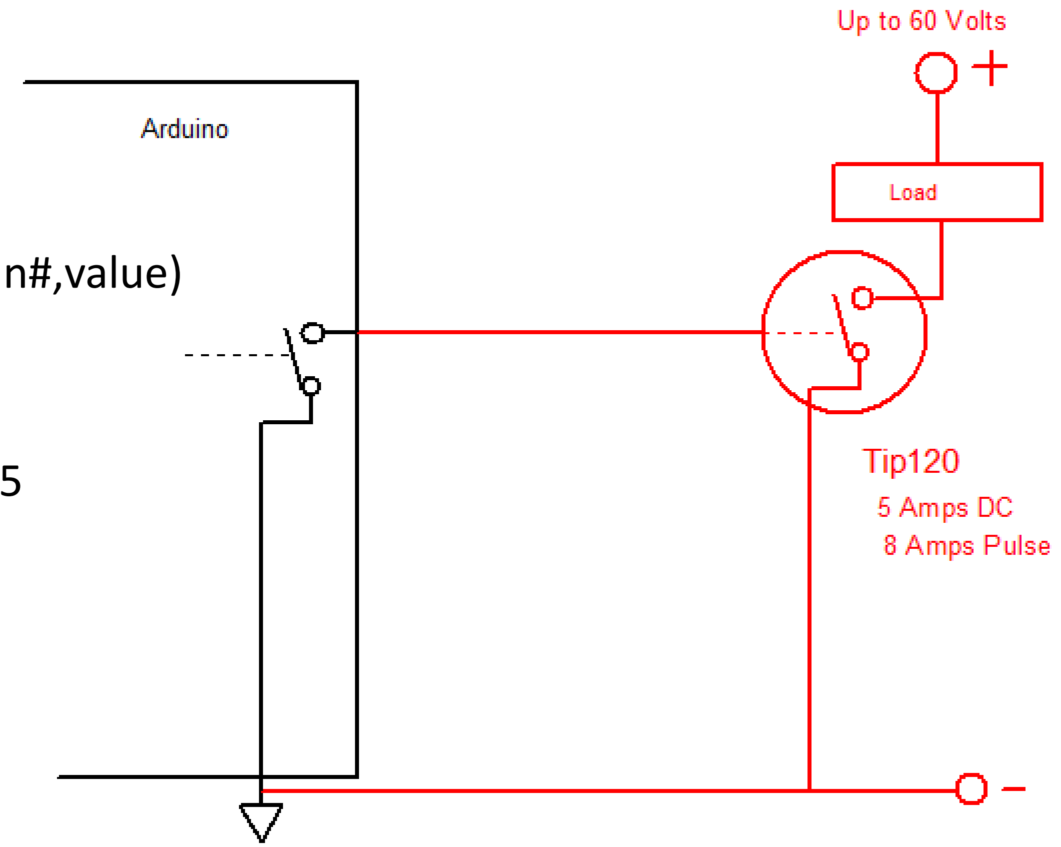
50% of full speed

75% of full speed

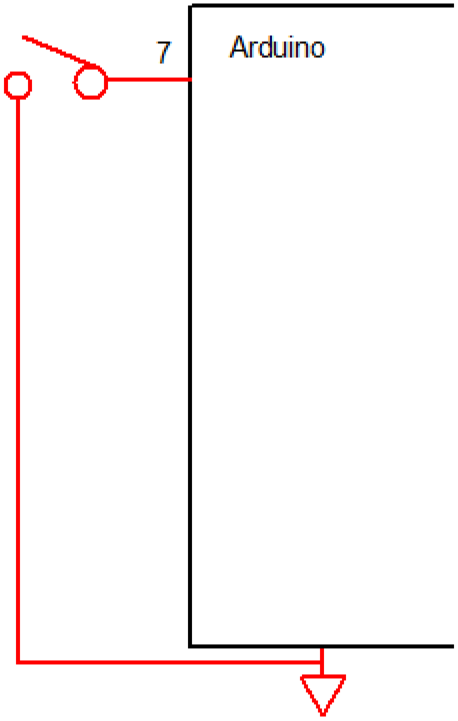
100% of full speed

AnalogWrite(pin#,value)

Value = 0 to 255

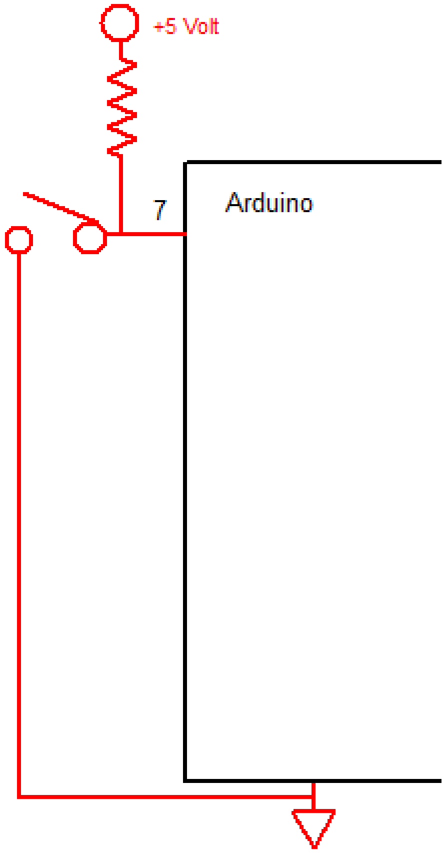


Simple input

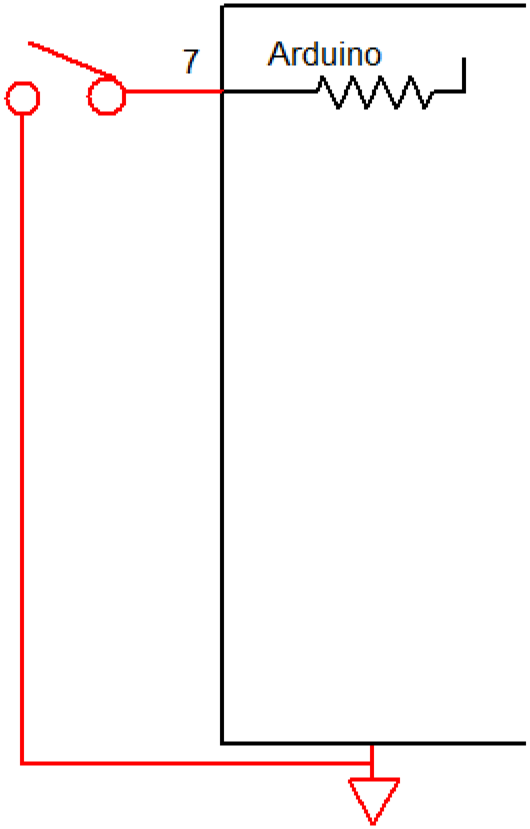


```
void setup() {
        pinMode(7, INPUT); // sets the digital pin 7 as input }
void loop() {
        val = digitalRead(7); // read the input pin
}
```

Simple input with Pull up resistor



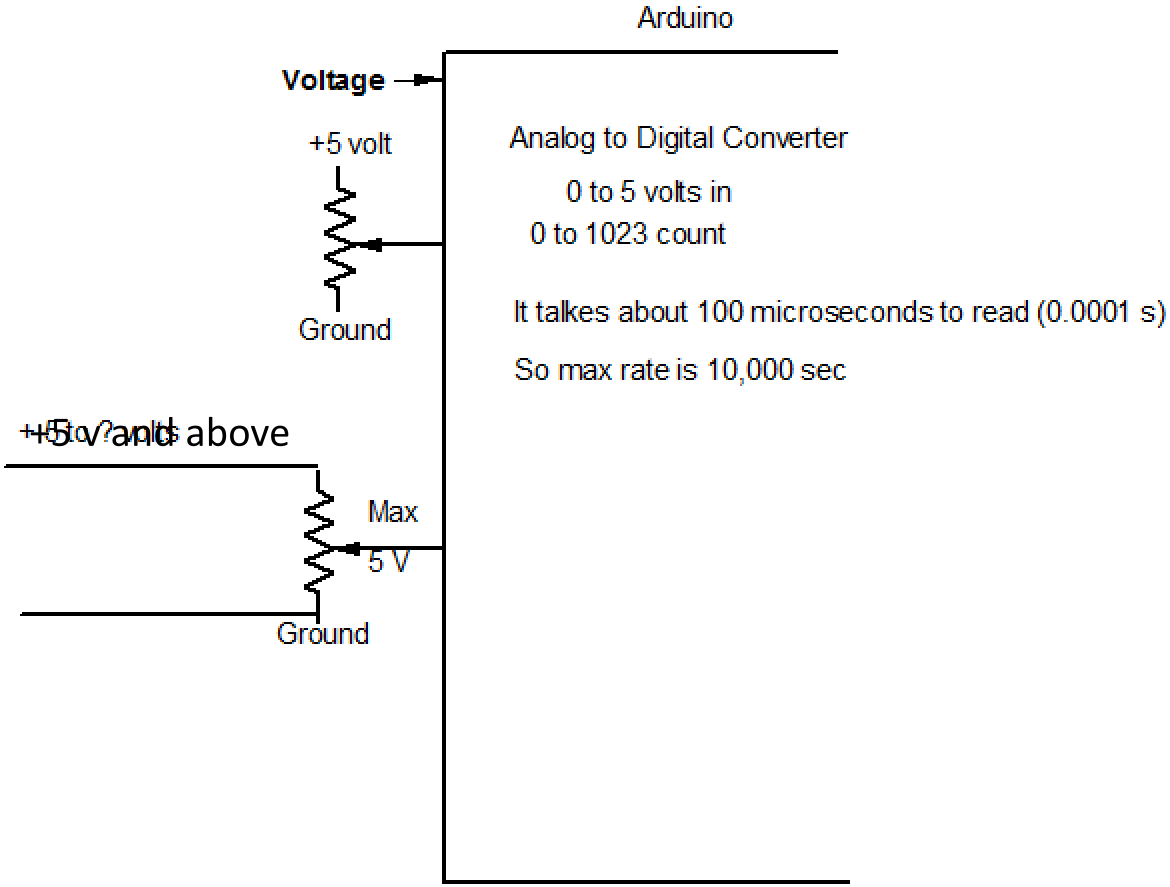
Using the microcontrollers internal pull up resistor



```
void setup()  {
                pinMode(7, INPUT); // sets the digital pin 7
                                                as input }
                digitalWrite(7, HIGH);  // set pull up resistor
                                                on
void loop() {
                val = digitalRead(7); // read the input pin
}
```

Sketch using switch input and light output

```
Int val        // define val as an integer variable

void setup()  {
          pinMode(7, INPUT);      // sets the digital pin 7 as input }
          digitalWrite(7, HIGH);    // set pull up resistor on

          pinMode(10,OUTPUT);    //sets pin 10 as output
}

void loop() {
          val = digitalRead(7);        // read the input pin
          if (val == HIGH) {
                    digitalWrite(10,HIGH);
          } else {
                    digitalWrite(10,LOW);
          }
}
```



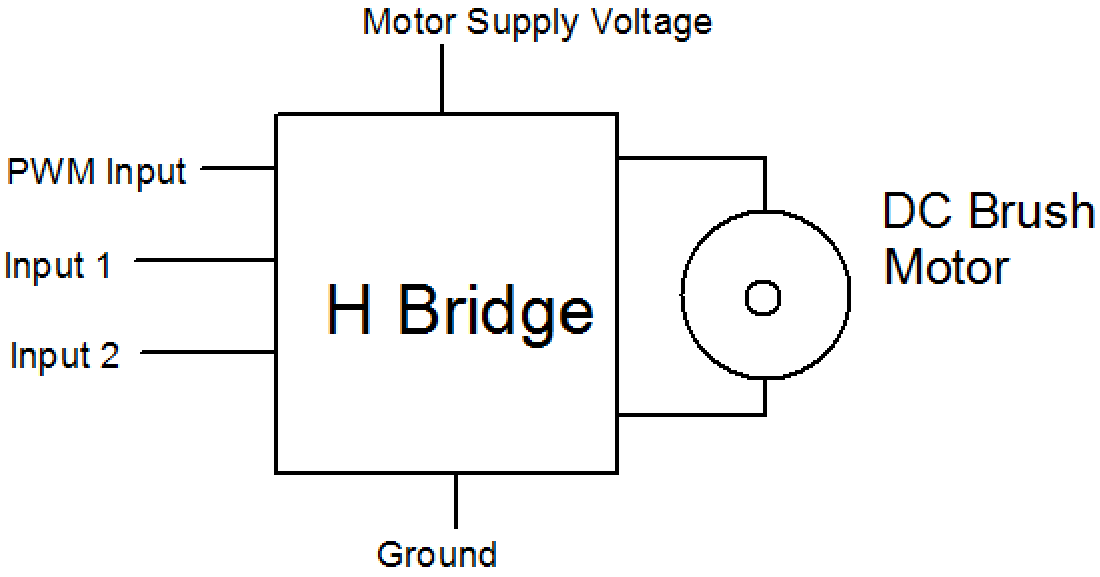
Analog to Digital Converter

0 to 5 volts in
0 to 1023 count

It talkes about 100 microseconds to read (0.0001 s)

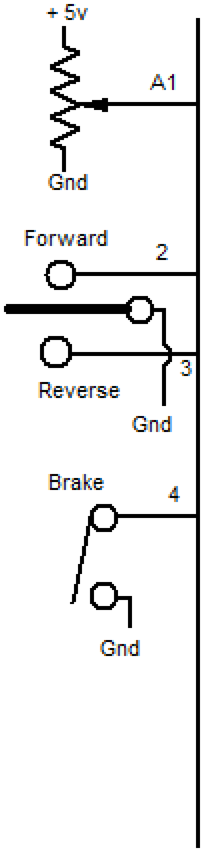
So max rate is 10,000 sec

Variable voltage input

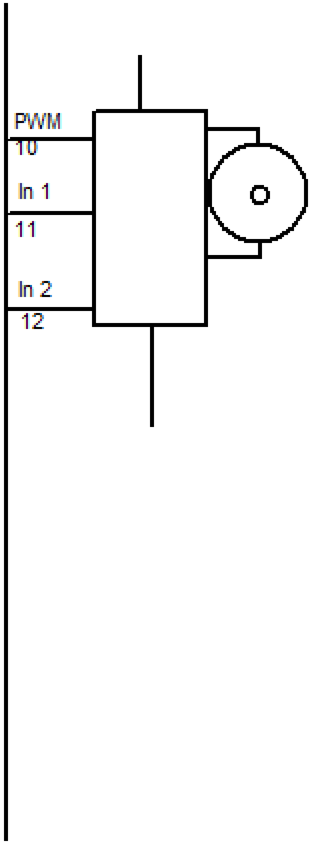


Input 1 Low	Input 2 Low	Motor freewheels
Input 1 High	Input 2 Low	Motor Forward
Input 1 Low	Input 2 High	Motor Reverse
Input 1 High	Input 2 High	Motor Brakes

Motor control using an H Bridge



```
Int speed;
Void setup {
        pinMode (A1,INPUT);
        pinMode(2, INPUT);
        digitalWrite(2, HIGH);
        pinMode(3, INPUT);
        digitalWrite(3, HIGH);
        pinMode(4, INPUT);
        digitalWrite(4, HIGH);

        pinMode(10, OUTPUT);
        pinMode(11,OUTPUT);
        pinMode(12,OUTPUT);
}

Void loop {
        speed = analogread(A1);    // values 0 to 1023
        analogwrite(10,speed/4);    //values from 0 to 255
        if (digitalRead(2)== HIGH) {
                digitalWrite(11, HIGH);
                digitalWrite(12,LOW);
        }
        if(digitalRead(3)==HIGH) {
                digitalWrite(11,LOW);
                digitalWrite(12,HIGH);
        }
        if(digitalRead(4)==HIGH){
                digitalWrite(11,HIGH);
                digitalWrite(12,HIGH);
        }
}
```



Shields

Special purpose boards to plug into a base Arduino.

- Stepper Driver
- Motor Driver
- Ethernet Driver
- Blue Tooth
- Prototyping
- RC Servo
- Cellular
- Color LCD
- Many others

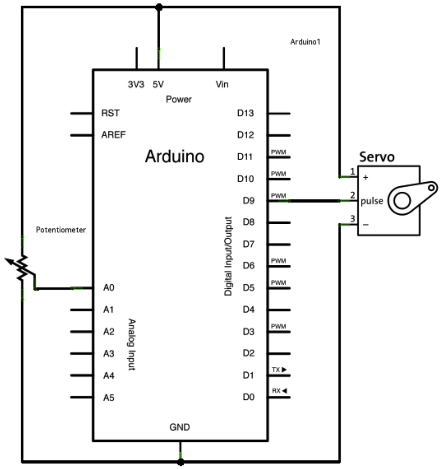
Libraries

- Groups of functions developed by individuals.
- Used to simplify complex operations

Examples:

- Motor
- Stepper
- Serial Communications
- RC Servo
- Keypad
- LCD Display
- Ethernet
- GPS

Controlling a RC servo position using a potentiometer



```
#include <Servo.h>

Servo myservo;        // create servo object to control a servo

int val;              // variable to read the value from the analog pin

void setup() {
     myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
    val = analogRead(A0);              // reads the value of the potentiometer
                                      // (value between 0 and  1023)
   val = map(val, 0, 1023, 0, 179);   // scale it to use it with the servo
                                      // (value between 0 and 180)
   myservo.write(val);               // sets the servo position according to the scaled value
   delay(15);                        // waits for the servo to get there
}
```

Growth Path from base Arduino

Hardware: Arduino Mega 2560
ARM series of microcontrollers

Software: AVR Studio

Sources

Local Hardware

- Radio Shack
- Microcenter (Book section)

Online Hardwire

- www.sparkfun.com
- www.adafruit.com
- www.pololu.com

Online Software and Information

- www.arduino.cc
- Google Arduino and any device