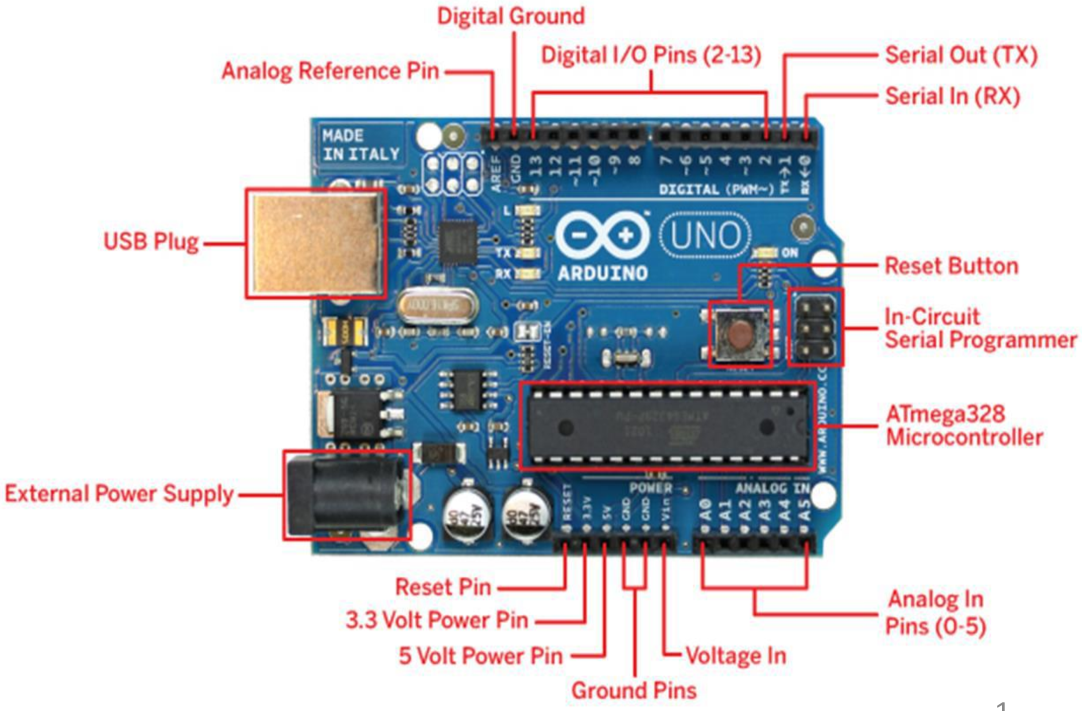
The Arduino Uno Development Board

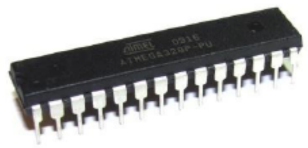
- Development boards are circuit boards designed to facilitate work with a particular microcontroller.
- The **Arduino Uno** is an **open source** development board which uses the **ATmega328** microcontroller.
- Arduino is Italian meaning “Strong friend”
- The Arduino Uno board includes
  - ➢ 12V input power supply
  - ➢ USB plug for communication and programming.
  - ➢ Reset button
  - ➢ Input/Output connectors
  - ➢ Crystal oscillator (16Mhz)

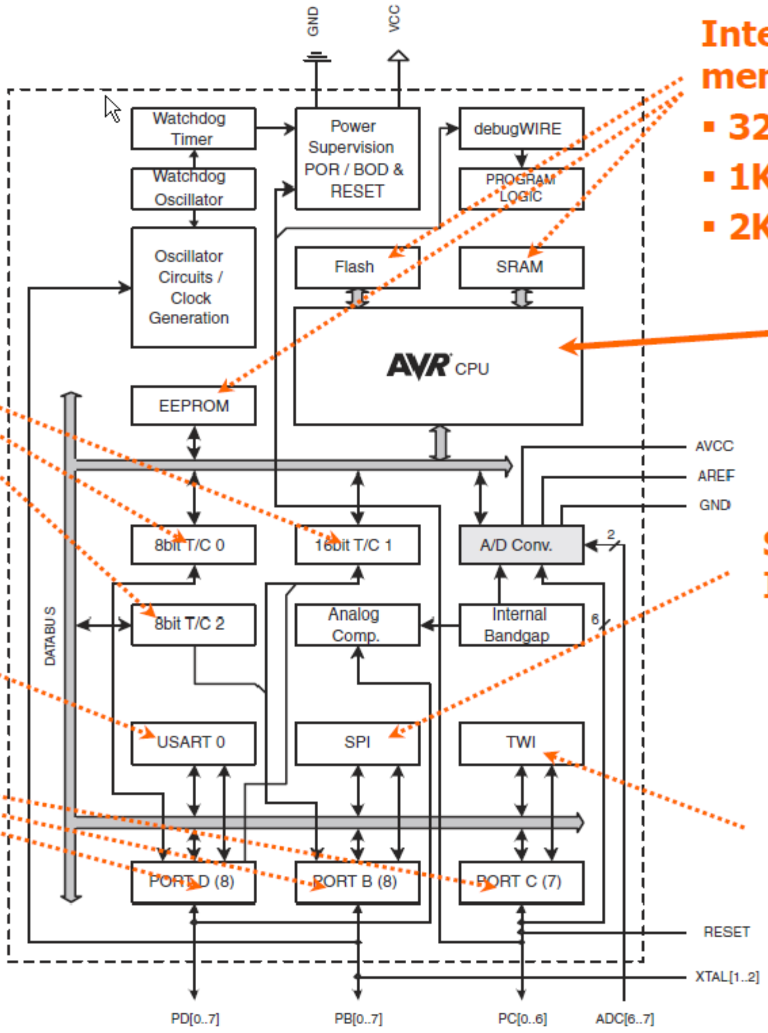


The ATmega328p

- The ATmega328P is a low-power 28 pin CMOS **8-bit** microcontroller based on the AVR® enhanced **RISC** architecture.
- The AVR also uses **Harvard** architecture to maximize performance.
- Used on the Arduino Uno board
- Some of its features include
  - 131 Powerful Instructions
  - 32 x 8 General Purpose Working Registers
  - 32KBytes of In-System Self-Programmable Flash program Memory
  - 1KBytes EEPROM
  - Two 8-bit Timer/Counters with Separate Prescaler
  - One 16-bit Timer/Counter with Separate Prescaler
  - 6-channel 10-bit ADC (PDIP Package)
  - One On-chip Analog Comparator
  - Power-on Reset and Programmable Brown-out Detection
  - External and Internal Interrupt Sources

ATmega328p Overview





Internal memories

- **32KB Flash**
- **1KB EEPROM**
- **2KB SRAM**

8-bit CPU

Timer/Counter

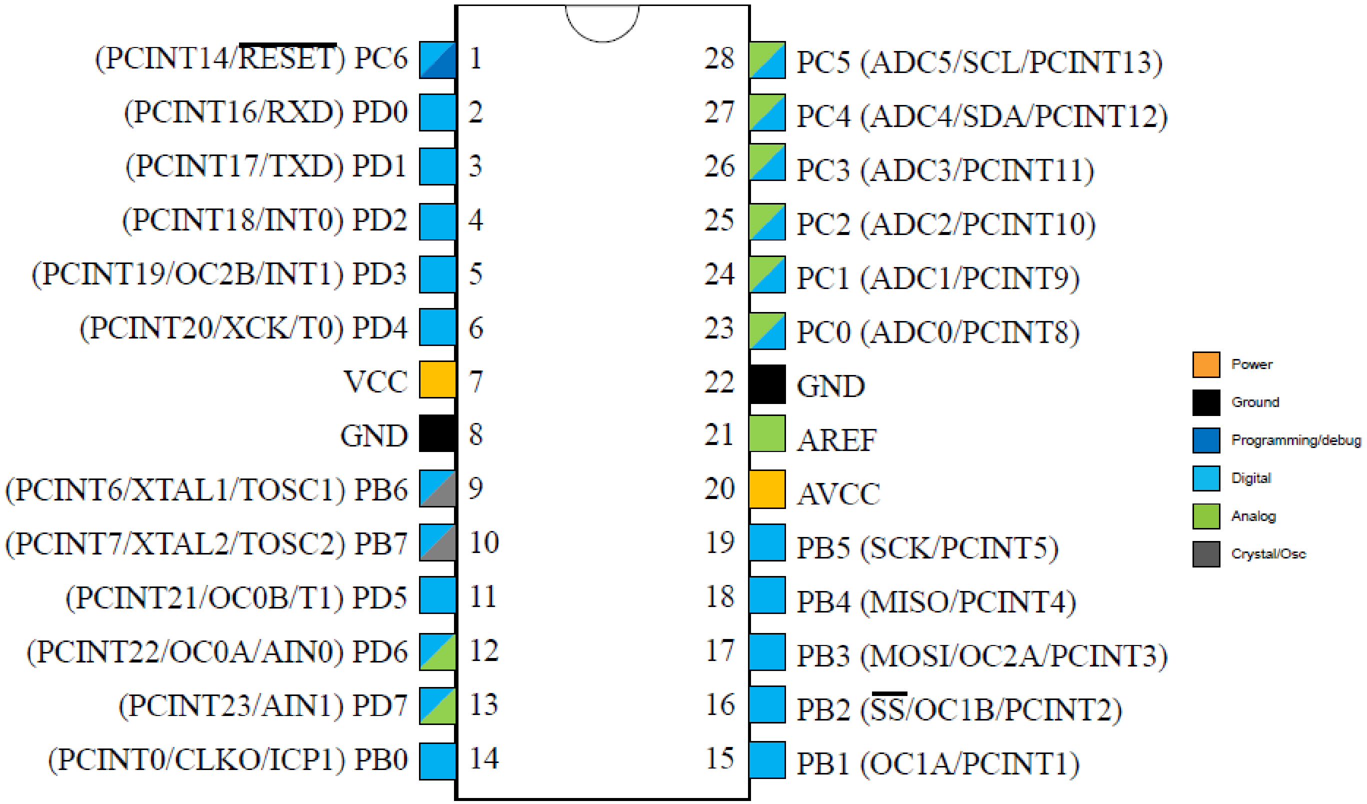
Serial Peripheral Interface

Universal Synchronous and Asynchronous serial Receiver and Transmitter (Serial)

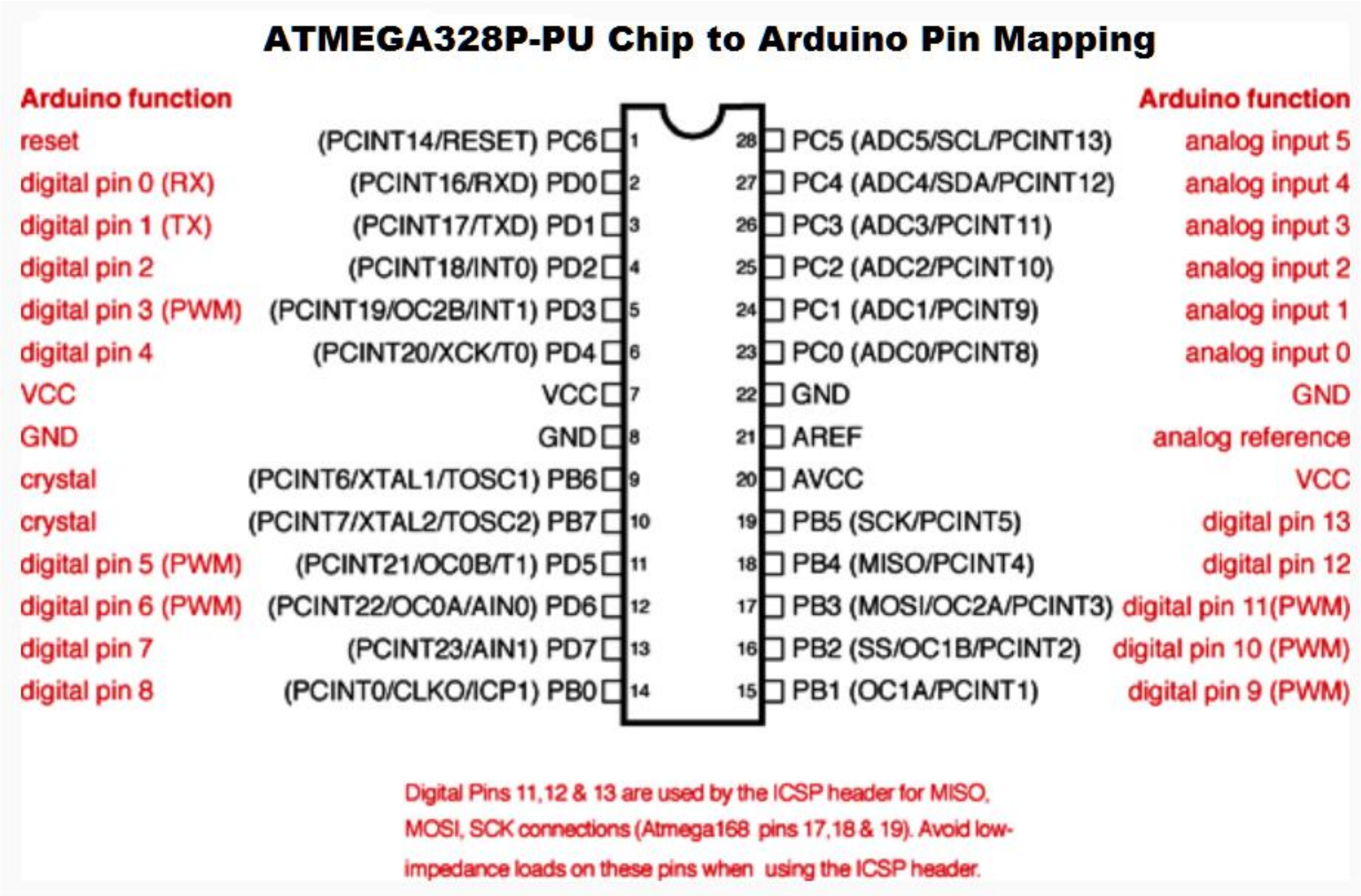
GPIO

2-wire Serial Interface

ATmega328p Pin Assignment



Arduino Uno ATmega328p Pin Mapping



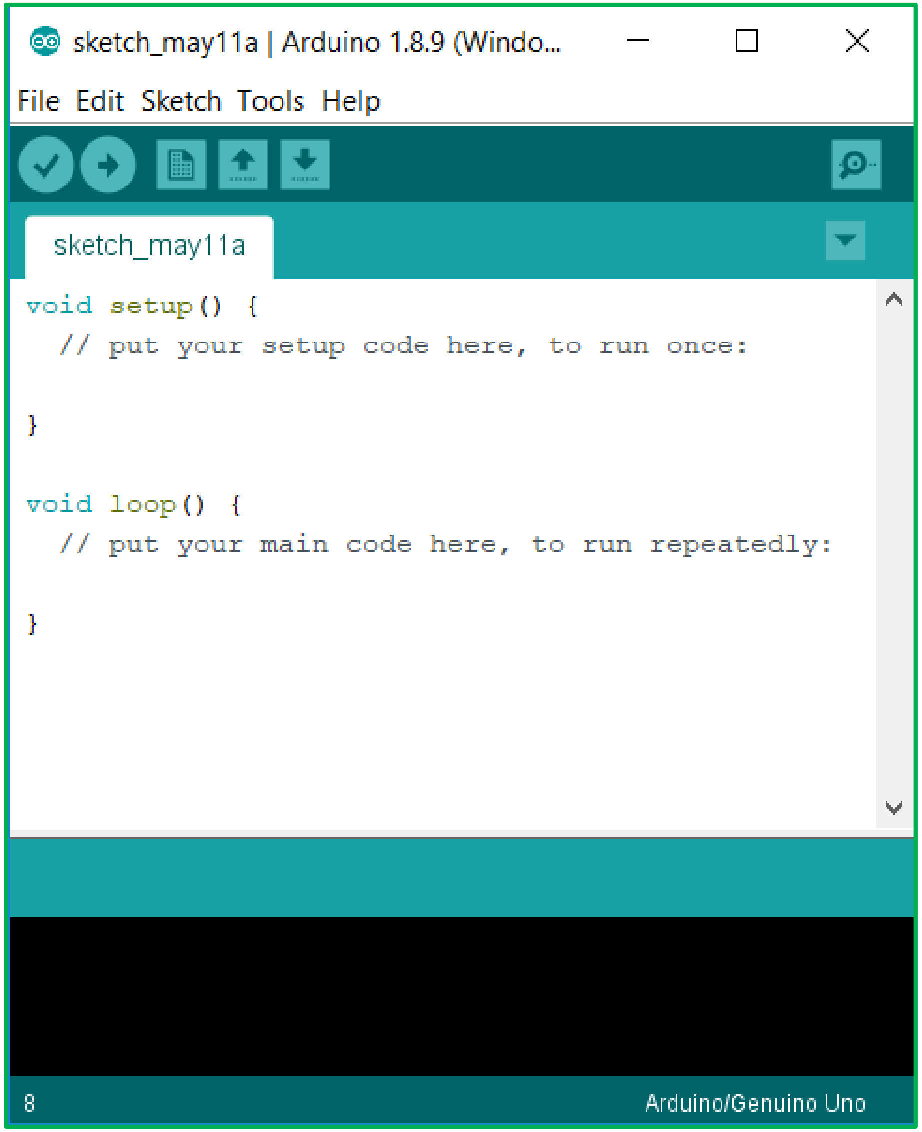
ATMEGA328P-PU Chip to Arduino Pin Mapping

Arduino function					Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11,12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Arduino Uno IDE

- The Arduino Uno also has a **free IDE** for **writing** and **uploading** program code unto the board’s microcontroller.
- The Aduino IDE is based on the **C language**.
- A program in the Arduino IDE is referred to as a ***Sketch***.
- Commands in the IDE Sketches are **case** sensitive.
- A Sketch may be compiled and loaded into the Arduino chip from the ***Sketch*** dropdown **menu**.
- Alternatively, a sketch may also be compiled and **exported** to other platforms using ‘**Export compiled Binary**’ also in the *Sketch* dropdown menu.


sketch_may11a | Arduino 1.8.9 (Windo...
File Edit Sketch Tools Help
sketch_may11a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Arduino IDE

Arduino IDE Example: Blinking LED

- The following code blinks an LED on Pin 13 of the Arduino Uno board

```
  // This program flashes an LED on Digital Output pin 13 //
  of the Arduino board which corresponds to PortB.5 of //
  the ATmega328P chip.

  void setup() {
    // put your setup code here, to run once:
  pinMode(13, OUTPUT);
  }

  void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(13, HIGH);     //LED ON
    delay(500);                 //Delay for 500 ms
    digitalWrite(13, LOW);      //LED OFF
    delay(500);                 //Delay 500 ms
}
```

- A semicolon is used to indicate the end of a statement in the Arduino IDE

Arduino IDE Libraries

- **Libraries** are a collection of code that simplify interfacing to a sensor, display, module etc.
- The Arduino IDE has many built in libraries. For instance the **LiquidCrystal** library makes it easy to communicate with a character LCD.
- There are also **numerous** libraries that can be **downloaded** online like the HCSR04 ultrasonic sensor.
- The downloaded library can be imported into the Arduino Ide using **Include Library** in the **Sketch** dropdown menu.
- *The **#include*** keyword is used to include libraries in an Arduino Sketch.
- The following example uses the **serial terminal** of the Arduino IDE to displays the distance in cm as measured from the **HCSR04** sonar sensor.

```
#include <HCSR04.h>

UltraSonicDistanceSensor distanceSensor(13, 12);
 // Initialize sensor that uses digital pins 13 and 12.

void setup () {
   Serial.begin(9600);  // initialize serial connection s
}

void loop () {
   // do a measurement using the sensor in centimeters.
   Serial.println(distanceSensor.measureDistanceCm());
   delay(500);
}
```

Arduino IDE ADC

- The following code reads analog input from the potentiometer and displays the output on an LCD in volts.

```
#include <LiquidCrystal.h>    //LCD library
// initialize the library by associating the LCD interface pins
// with the appropriate I/O pins.
const int rs = 0, en = 1, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
  int sensorPin = A0;    // select the input pin for the potentiometer
  int sensorValue = 0;  // variable to store the value coming from the sensor
  float senCal = 0.0048876;
  float volts = 0.0;

void setup() {
  lcd.begin(16, 2); // set up the LCD's number of columns and rows:
  lcd.clear();       // clear the LCD
  lcd.print("Arduino ADC"); // Print a message to the LCD.

}
void loop() {
  lcd.clear();       // clear the LCD
  lcd.print("Arduino ADC"); // Print a message to the LCD.
  sensorValue = analogRead(sensorPin);  //acquire reading from ADC
  lcd.setCursor(1,2);         //locate the 1st col of 2nd row of LCD
  lcd.print(sensorValue);     //display the 10bit sensor reading
  lcd.print(" ==>");
  volts = sensorValue*senCal; //calculate the actual voltage
  lcd.setCursor(9,2);         //move the cursor to col, row
  lcd.print (volts);          //display the computed voltage
  lcd.print ("V");            //append the SI unit to the result
  delay(1000);
}
```