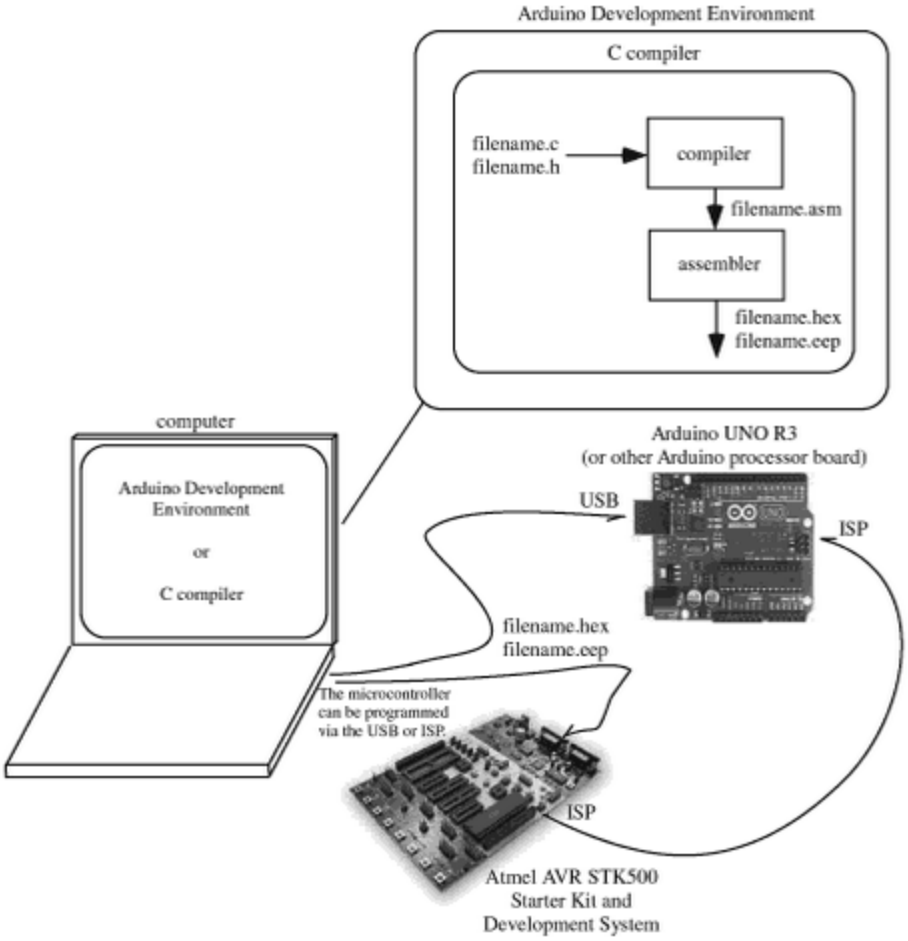


Figure 2.1: Programming the Arduino processor board. (Used with permission from SparkFun Electronics (CC BY–NC–SA), and Atmel, Incorporated.)

Anatomy of a Program

Programs written for a microcontroller have a fairly repeatable format.
Slight variations exist but many follow the format provided.

```
// Comments containing program information
// - file name
// - author & date
// - revision history (author, date, brief description of modifications)
// - compiler setting information
// - hardware connection description to microcontroller pins
// - program description

 // Include Files
 #include<file._name.h> .

// Function Prototypes
 A list of functions and their format used within the program

// Program Constants
 #define TRUE 1
 #define FALSE0
 *define ON 1
 #define OFF 0

 // Interrupt Handler Definitions
 Used to link the software to hardware interrupt features

// Global Variables
 Listing of variables used throughout the program

// Main Program
     void setup( )
     {
     }

     void loop( )
     {
     }

// Function Definitions
 A detailed function definition for each function used within the program.
```

Arduino Language Reference

Arduino programs can be divided in three main parts: structure, values (variables and constants), and functions.

Structure

Structure
setup( )
loop( )

Additional Syntax
; (semicolon)
{} (curly braces)
// (single line comment)
/* */ (multi-line comment)
#define
#include

Control Structures
if
if...else
for
switch case
while
do... while
break
continue
return
goto

Arithmetic Operators
= (assignment operator)
+ (addition)
- (subtraction)
* (multiplication)
/ (division)
% (modulo)

Comparison Operators
= =(equal to)
!= (not equal to)
< (less than)
> (greater than)
<= (less than or equal to)
>= (greater than or equal to)
Boolean Operators
&& (and)
|| (or)
! (not)

Pointer Access Operators
* dereference operator
& reference operator

Bitwise Operators
& (bitwise and)
| (bitwise or)
^ (bitwise xor)
~ (bitwise not)
<< (bitshift left)
>> (bitshift right)
Compound Operators
++ (increment)
-- (decrement)
+= (compound addition)
-= (compound subtraction)
*= (compound multiplication)
/= (compound division)
&= (compound bitwise and)
|= (compound bitwise or)

Arduino Language Reference

Values (Variables and Constrants)

Constants

HIGH | LOW
INPUT | OUTPUT | INPUT_PULLUP
LED_BUILTIN
true | false
integer constants
floating point constants

Data Types

void
boolean
char
unsigned char
byte
int
unsigned int
word
long
unsigned long
short
float
double
string - char array
String - object
array
Conversion
char( )
byte( )
int( )
word( )
long( )
float( )

Variable Scope & Qualifiers

variable scope
static
volatile
const
Utilities
sizeof( )

Utilities

sizeof( )
PROGMEM

Arduino Language Reference

Commands/Functions

Digital I/O
pinMode( )
digitalWrite( )
digitalRead( )

Analog I/O
analogReference( )
analogRead( )
analogWrite( ) - PWM

Advanced I/O
tone( )
noTone( )
shiftOut( )
shiftIn( )
pulseIn( )
Time
millis( )
micros( )
delay( )
delayMicroseconds( )

Mathematical
min( )
max( )
abs( )
constrain( )
map( )
pow( )
sqrt( )

Trigonometric
sin( )
cos( )
tan( )

Random Numbers
randomSeed( )
random( )
Bits and Bytes
lowByte( )
highByte( )
bitRead( )
bitWrite( )
bitSet( )
bitClear( )
bit( )

Interrupts

External Interrupts
attachInterrupt( )
detachInterrupt( )

Interrupts
interrupts( )
noInterrupts( )

Communication
Serial
Stream

Arduino Functions

Digital I/O	Analog I/O	Advanced I/O	Time	Mathematical
pinMode( ) digitalWrite( ) digitalRead( )	analogReference( ) analogRead( ) analogWrite( )	Tone( ) noTone( ) shiftOut( ) shiftIn( ) pulseIn( )	millis( ) micros( ) delay( ) delayMicroseconds( )	min( ) max( ) abs( ) constrain( ) map( ) pow( ) sqrt( )
Communications	**Interrupts**	**Bits & Bytes**	**Random Numbers**	**Trigonometric**
Serial( ) Stream( )	**External Interrupts** attachInterrupt( ) detachInterrupt( ) **Interrupts** interrupts( ) noInterrupts( )	lowBytes( ) highBytes( ) bitRead( ) bitWrite( ) bitSet( ) bitClear( ) bit( )	randomSeed( ) random( )	sin( ) cos( ) tan( )

Arduino UNO R3 Interrupts (Interrupt Service Routine ISR)

Interrupt 0 1
Pins D2 D3

Mode	Operation (Triggers IRS)	Remarks
Low	Whenever trigger is Low	ISR runs continuously - rarely used
Rising	As trigger changes from Low to High	
Falling	As trigger changes from High to Low	
Change	Whenever trigger toggles	
High	Not Applicable to the UNO	

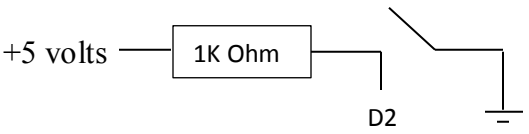
```
// Interrupt Service Routine
// doSomething when Interrupt 0 (pin D2 changes from High to Low
// Use internal pull-up 40 KOhm resistor for hard switch; not required for digital sensor trigger
// pinMode(2, INPUT_PULLUP);

void Setup( )
{
  attachInterrut(0, doSomething, Falling);
}

void Loop( )
{blah; blah; blah;}

doSomething( )
{etc; etc; etc;}
```

Note:
hard wired pull-up resistor
open switch terminal to ground
+ 5 volts to top of 1K ohm resistor to closed switch contact to Interrupt pin (D2 or D3)



Arduino UNO R3 I/O Pins http://playground.arduino.cc/Learning/Pins

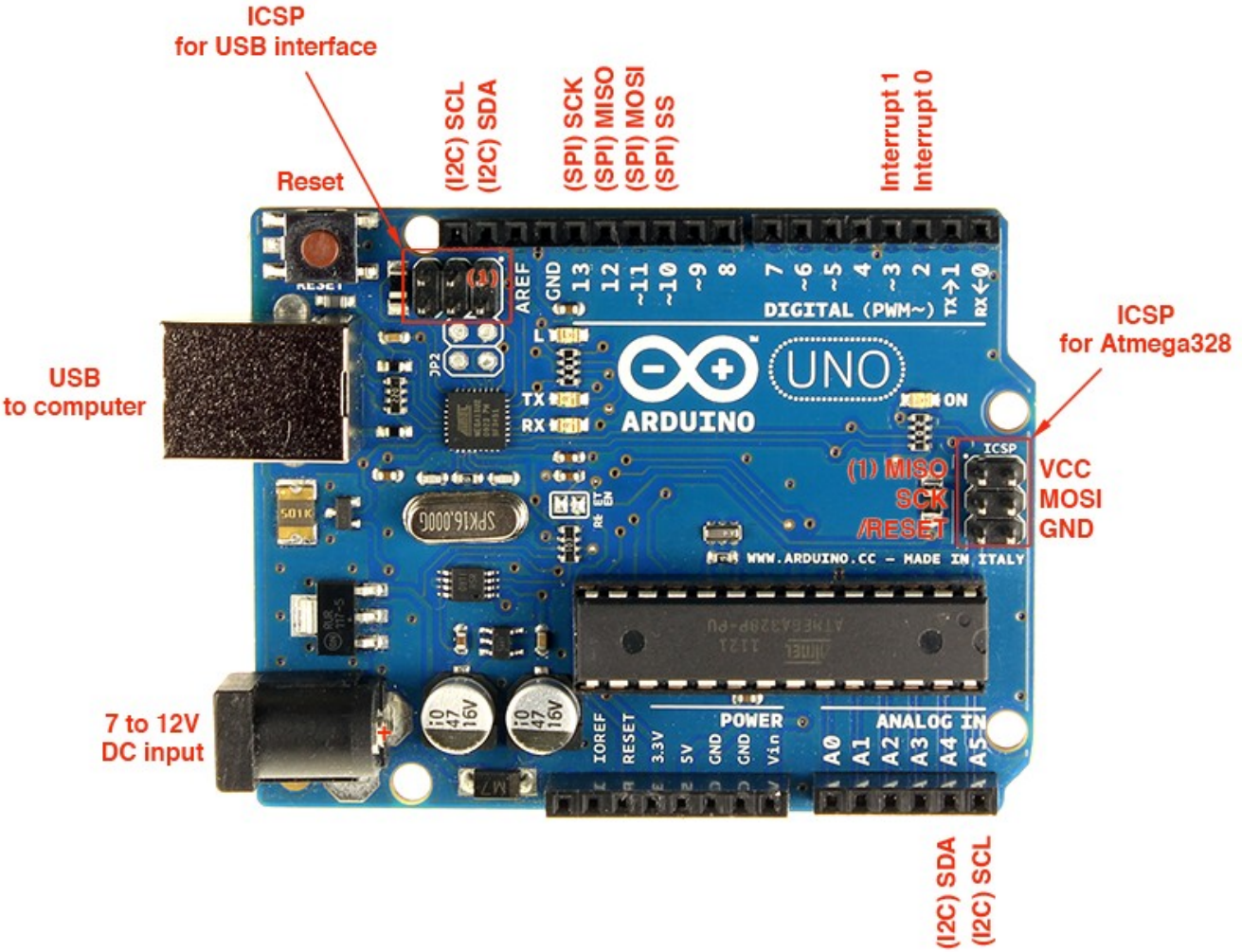
		I/O	PWM Duty Cycle	10 bit ADC	TWI / I^2C	High Speed	Triggers	USB / FTDI	USB / FTDI
Pin	Alias	pinMode() digitalWrite() digitalRead()	analog Write()	analog Read()	.send()	.transfer()	attach Interrupt()	Serial .print()	Serial .read()
0	RX	YES	-	-	-	-	-	-	YES
1	TX	YES	-	-	-	-	-	YES	-
2	IRQ0	YES	-	-	-	-	YES *	-	-
3	IRQ1	YES	YES (2)	-	-	-	YES *	-	-
4	-	YES	-	-	-	-	-	-	-
5	-	YES	YES (0)	-	-	-	-	-	-
6	-	YES	YES (0)	-	-	-	-	-	-
7	-	YES	-	-	-	-	-	-	-
8	-	YES	-	-	-	-	-	-	-
9	-	YES	YES (1)	-	-		-	-	-
10	SS	YES	YES (1)	-	-	YES **	-	-	-
11	MOSI	YES	YES (2)	-	-	YES **	-	-	-
12	MISO	YES	-	-	-	YES **	-	-	-
13	SCK	YES	-	-	-	YES **	-	-	-
14	A0	YES	-	YES *	-	-	-	-	-
15	A1	YES	-	YES *	-	-	-	-	-
16	A2	YES	-	YES *	-	-	-	-	-
17	A3	YES	-	YES *	-	-	-	-	-
18	A4	YES	-	YES *	YES **	-	-	-	-
19	A5	YES	-	YES *	YES **	-	-	-	-

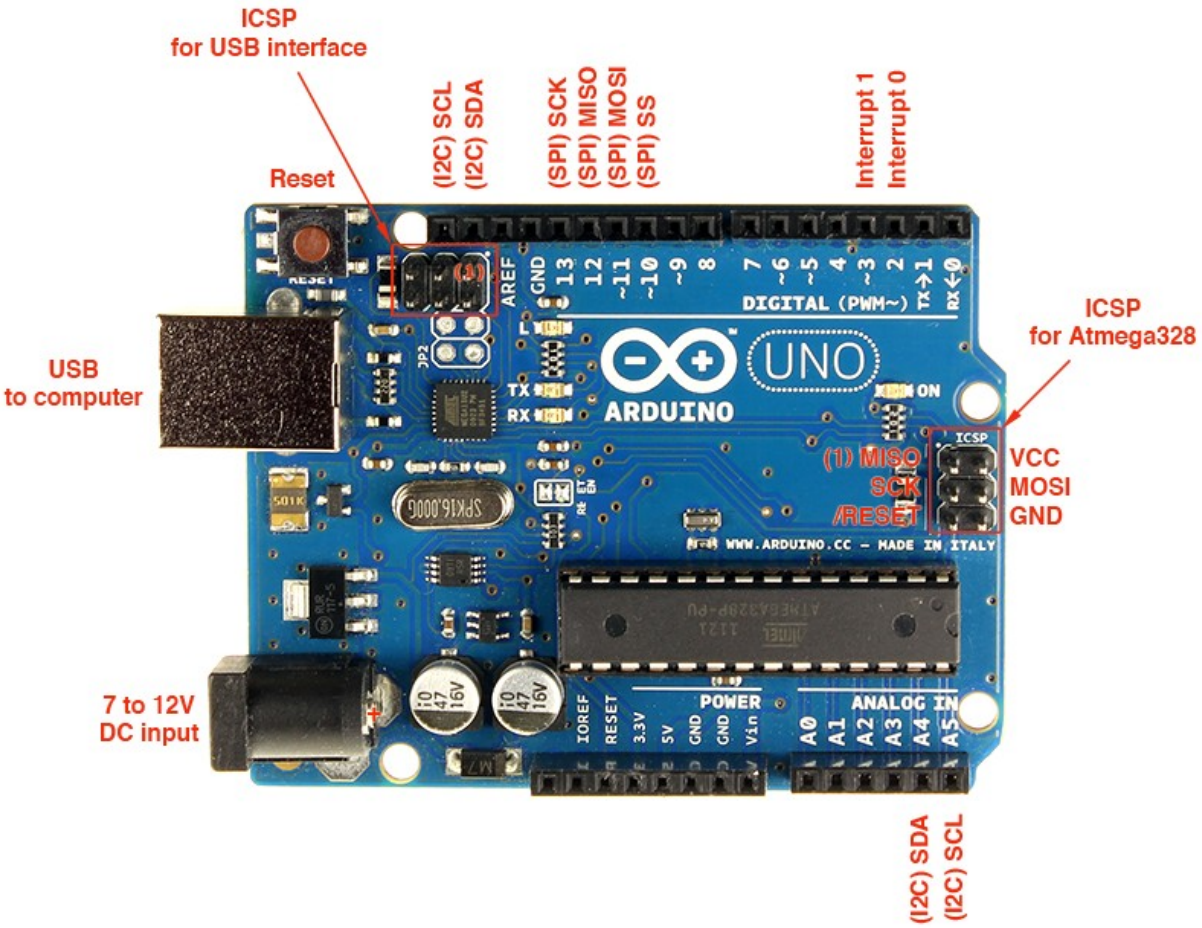
* The function expects the pin numbering scheme in the Alias column.
** Multiple pins must be used in conjunction for the specialized feature to work.
(#) Any PWM output is driven on Timer #'s frequency. The duty cycle is independent of other PWM outputs.

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.



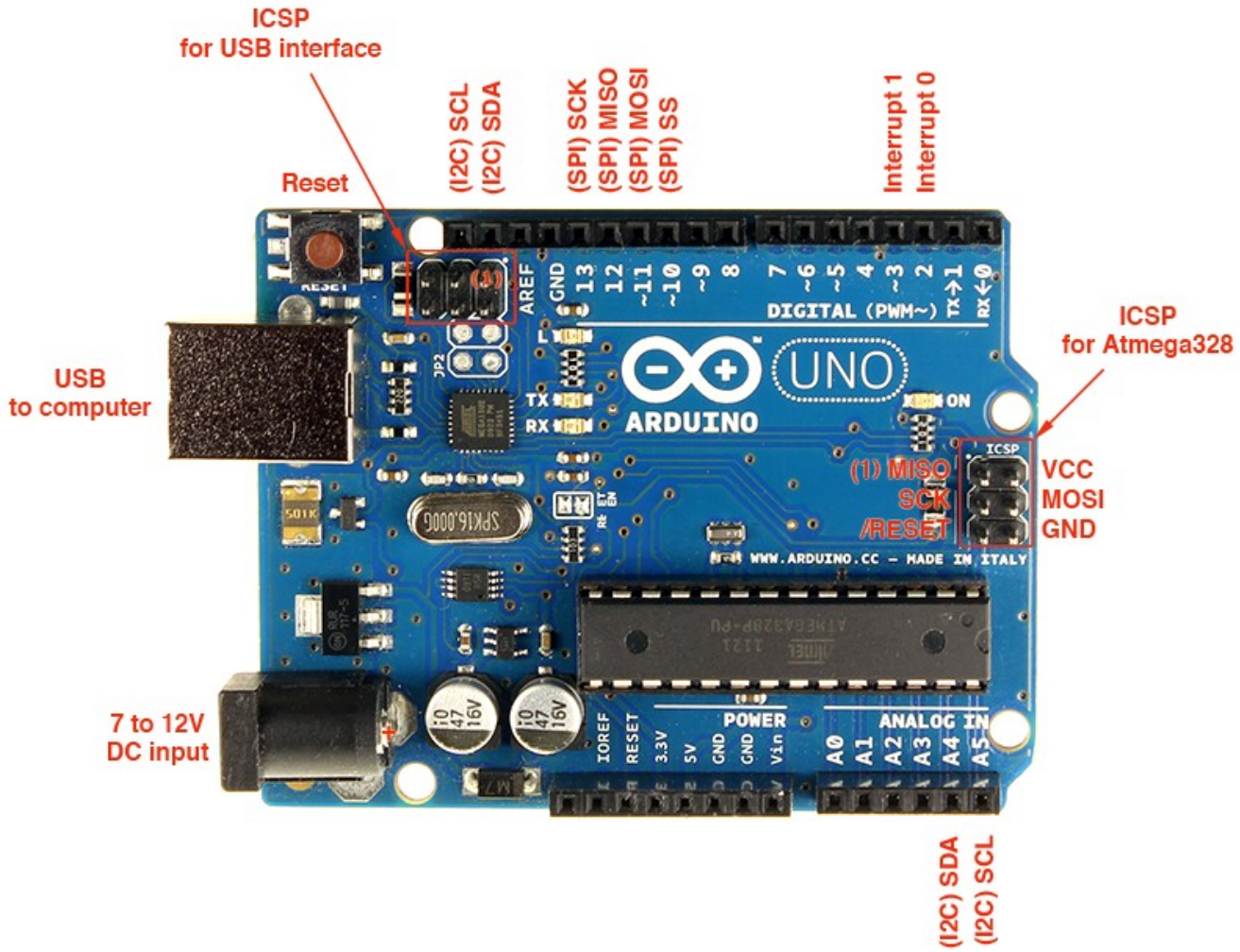


Top Row (10 Pin Header 14 Digital I/O) Left to Right

I2C (SCL)	Inter-Integrated Circuit Serial Bus (Serial Clock)			
I2C (SDA)	Inter-Integrated Circuit Serial Bus (Serial Data)			
AREF				
GND				
13	Digital	SPI (SCK)	Serial Peripheral Interface (Serial Clock)	LED
12	Digital	SPI (MISO)	Serial Peripheral Interface (Master In Slave Out)	
11 ~	Digital (PWM)	SPI (MOSI)	Serial Peripheral Interface (Master Out Slave In)	
10 ~	Digital (PWM)	SPI (SS)	Serial Peripheral Interface (Slave Select)	
9 ~	Digital (PWM)			
8	Digital			

Top Row (8 Pin Header) Left to Right

7	Digital	
6 ~	Digital (PWM)	
5 ~	Digital (PWM)	
4	Digital	
3 ~	Digital (PWM)	Interrupt 1
2	Digital	Interrupt 0
1	TX	
0	RX	



Bottom Row (8 Pin Header) Left to Right

NC
IOREF
RESET
3.3 V
5 V
GND
GND
Vin

Bottom Row (6 Pin Header 6 Analog I/O) Left to Right

A0	Analog In	
A1	Analog In	
A2	Analog In	
A3	Analog In	
A4	Analog In	TWI/I2C (SDA) Two-Wire Interface / Inter-Integrated Circuit Serial Bus (Serial Data)
A5	Analog In	TWI/I2C (SCL) Two-Wire Interface / Inter-Integrated Circuit Serial Bus (Serial Clock)

Arduino Input and Output

14 Digital I/O Pins

Each of the 14 digital pins on the UNO can be used as an input or output, using pinMode( ), digitalWrite( ), and digitalRead( ) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms.

In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt( ) function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite( ) function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

6 Analog I/O Pins

The UNO has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference( ) function.
Additionally, some pins have specialized functionality:

TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

Power Pins

The power pins are as follows:
Vin The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

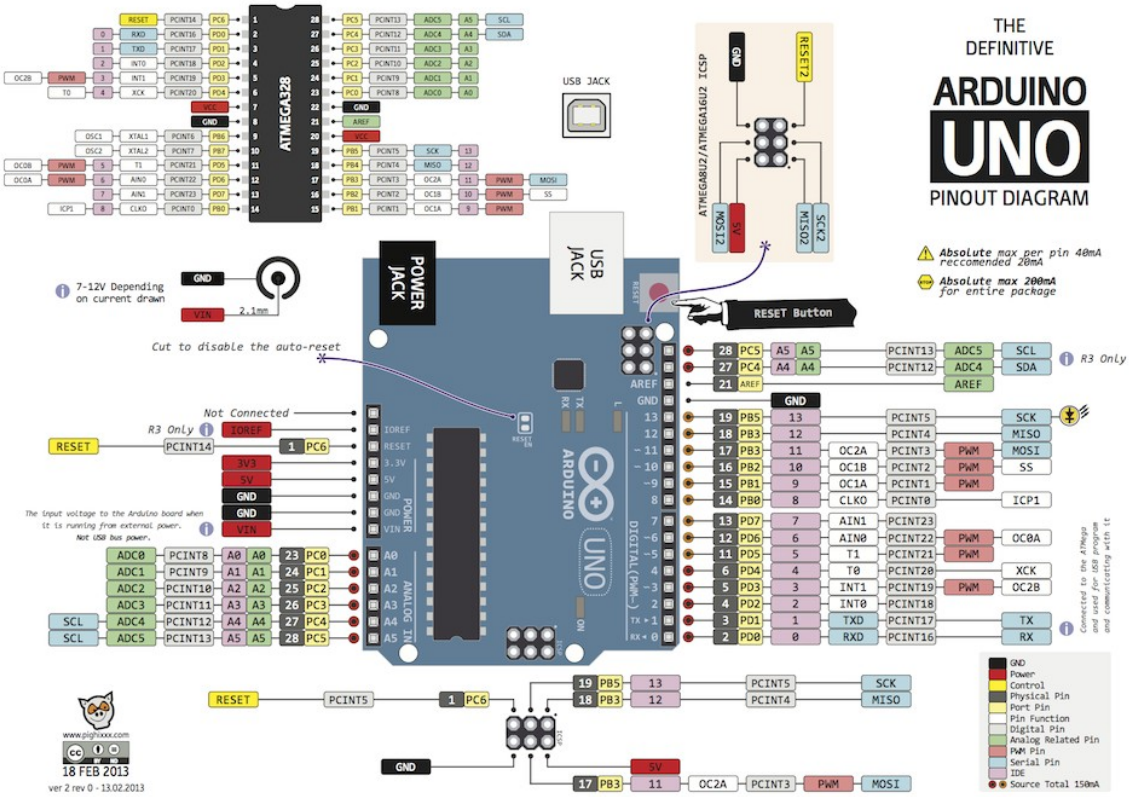
5V This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the Vin pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator.

3.3V volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND Ground pins.

IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V. AREF Reference voltage for the analog inputs. Used with analogReference( ).

Reset Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



Arduino UNO R3 Processing Board

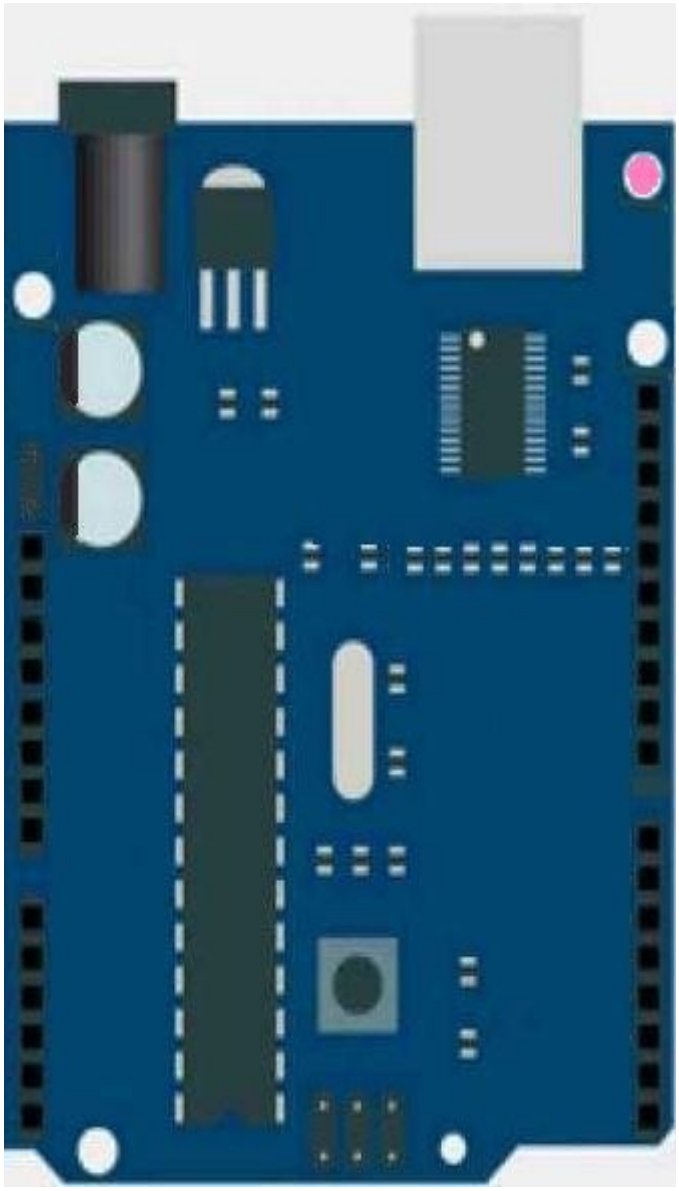
DC 7 to 12 Volt DC Power Supply Input
USB USART Connector for programming the processor via a host computer
USB to Serial Converter PC and the serial communications systems aboard the ATmega328 processor.
In System Programming (ISP) Connector
Reset Button Switch
TX LED & RX LED
Power Indicator LED & n On-Board Pin 13 LED
Voltage Regulator
16 MHz Clock

Header Strips
- 20 Sensors Input/Output I/O Pins
  - 14 Digital (including 5 with Pulse Width Modulation PMW and
    6 Analog Input (Analog-to-Digital (ADC) system) / Output Pins
- Miscellaneous Pins
  - External Power Supply Connector
  - Serial Communication (SPI, I2C)
  - Analog Reference Signal (AREF)
  - Input/Output Reference (IOREF
  - Board Voltage Supply
  - Reset
  - Ground

Inland Arduino UNO R3 Pin-Outs

X = No Header Pins for ProtoShield
~ = Pulse Width Modulation (PWM)
* D13 = On-Board LED
20 Digital IO Pins = 6 Analog + 14 Digital

On-Board LEDs
Power LED
Digital Output LED = D13
USB TX & RX LEDs



		X	I2C SCL
		X	I2C SDA
		AREF	
		GND	
X	NC	*D13	SPI SCK
X	IOREF	D12	SPI MISO
	Reset	~D11	SPI MOSI
	3.3 V	~D10	SPI SS
	5.0 V	~D9	
	GND	D8	
	GND		
	Vin	D7	
		~D6	
	A0	~D5	
	A1	D4	
	A2	~D3	Interrupt 1
	A3	D2	Interrupt 0
I2C SDA	A4	D1	TX
I2C SCL	A5	D0	RX

Inter-Integrated Circuit

I2C SCL	Synchronized Clock
I2C SDA	Synchronized Data

Serial Peripheral Interface

SPI SCK	Clock
SPI MISO	Master Input - Slave Output
SPI MOSI	Master Output - Slave Input
SPI SS	Slave Select

ProtoShield

Reset Switch
LED1 & LED2 (Anode)
Switch1 Normally Open Closed = Grounded
5 Pin GND Header
5 Pin +5 V Header