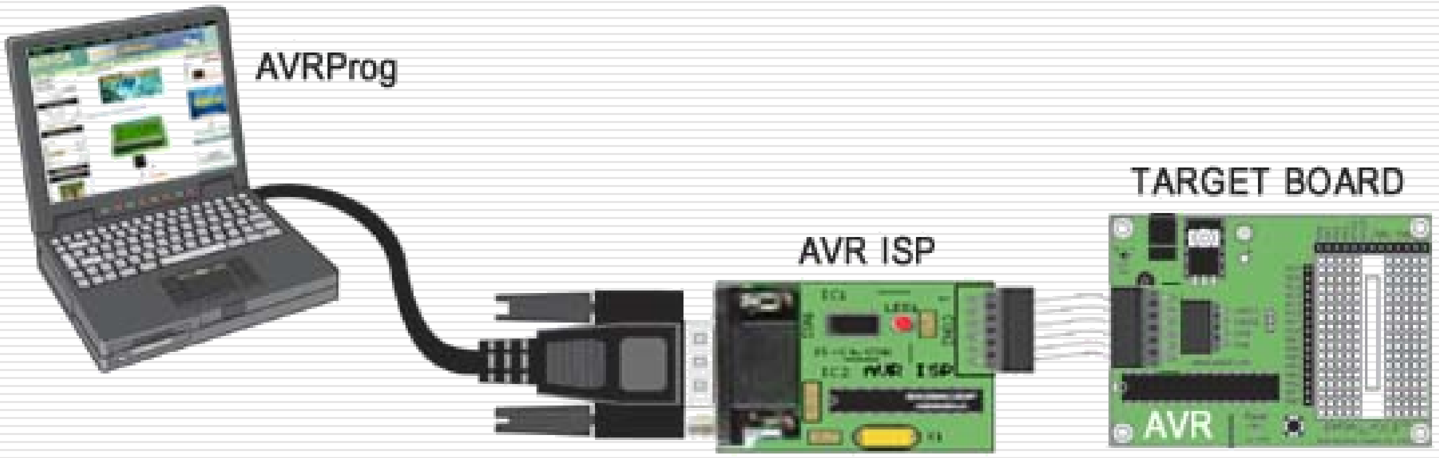
Programming sketches

Embedded Programming with Arduino Systems

Atmel programming

- Atmel processors are usually programmed via the ISP (In-circuit programmer) and dedicated programmer.



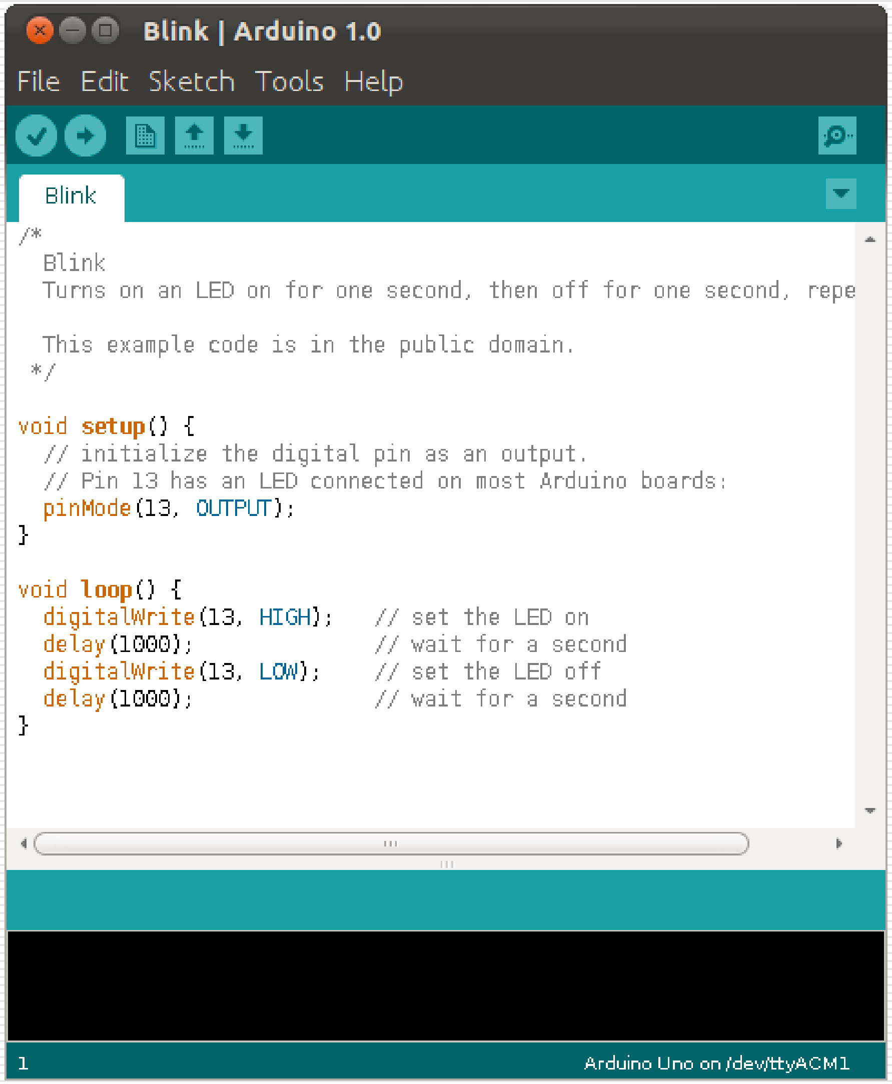
Ref: Electronics DIY – AVR Programmer

Programming with Arduino IDE & UNO

- The Uno uses a bootloader (2K code) which allows programming through the USB or FTDI interface.
  - ❑ When processor starts up
  - ❑ Loads and runs bootloader
  - ❑ If there is a programming command from the serial interface (USB or FTDI)
  - ❑ Loads the program that you are sending via USB/FTDI
  - ❑ Else runs the last loaded program.
- Bootloader and USB interface makes your work so much easier.

Using the Arduino IDE

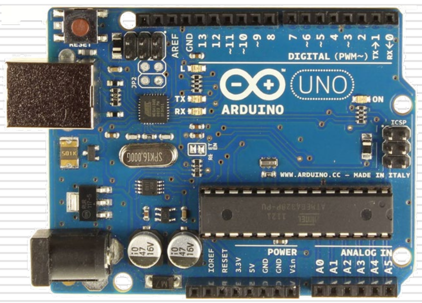
- Write your code
- Compile
- Upload to UNO board
- Press RESET button
- Observe results

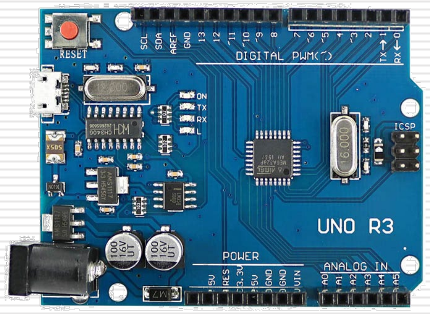


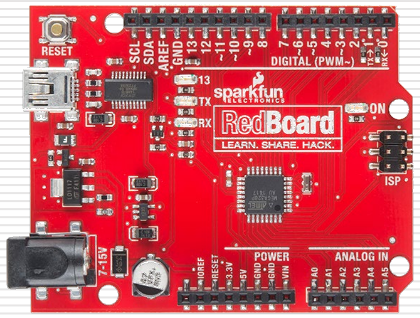
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repe

  This example code is in the public domain.
 */

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
}
```

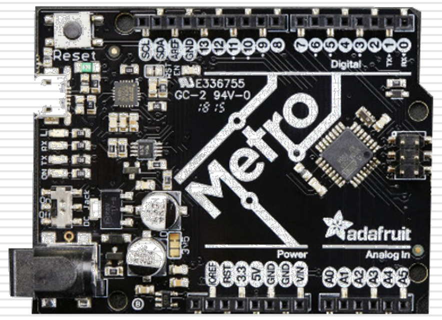
Variations of the UNO

- Being open-source, there are many variations.
- Programming and usage are basically the same with some minor variations.
- All boards use the ATMega328P processor (may be in different formats)
- All boards have the same I/O pins
- Difference is in $$cost$$



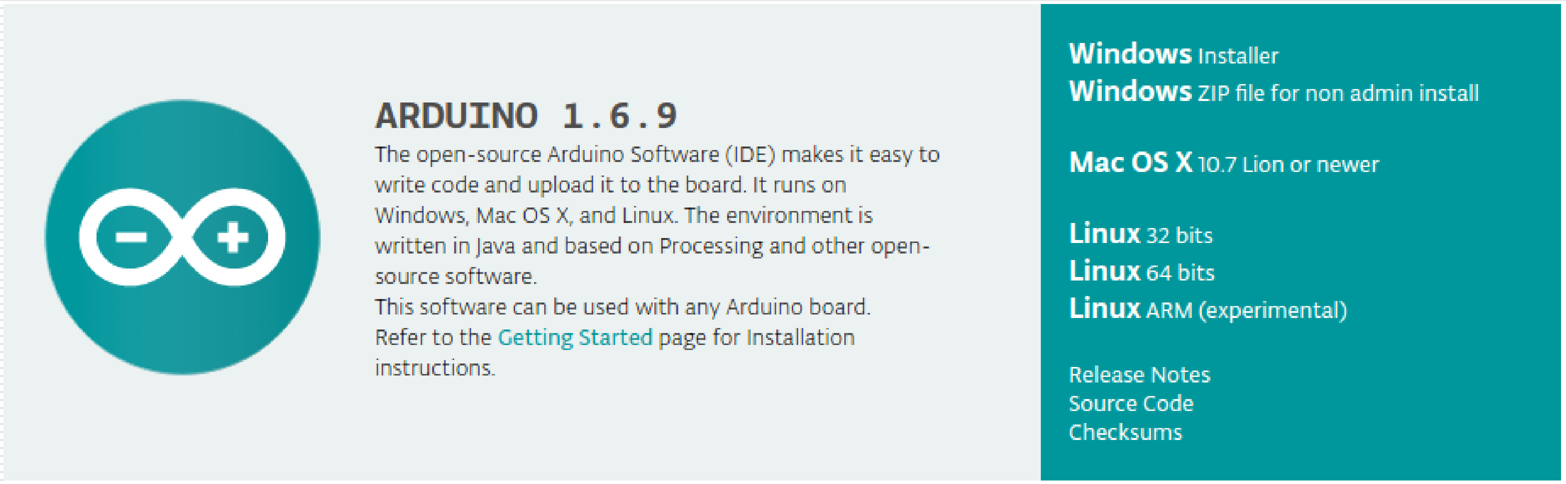






Arduino IDE Software

- Download and install the latest versions from the Arduino site.
- Current version 1.8.10
- Available in different platforms
- Copious help and how-tos available with simple search



ARDUINO 1.6.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

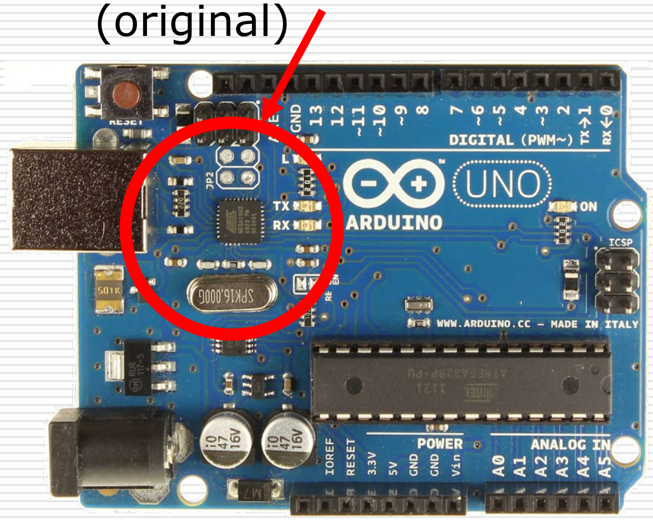
Linux 32 bits
Linux 64 bits
Linux ARM (experimental)

Release Notes
Source Code
Checksums

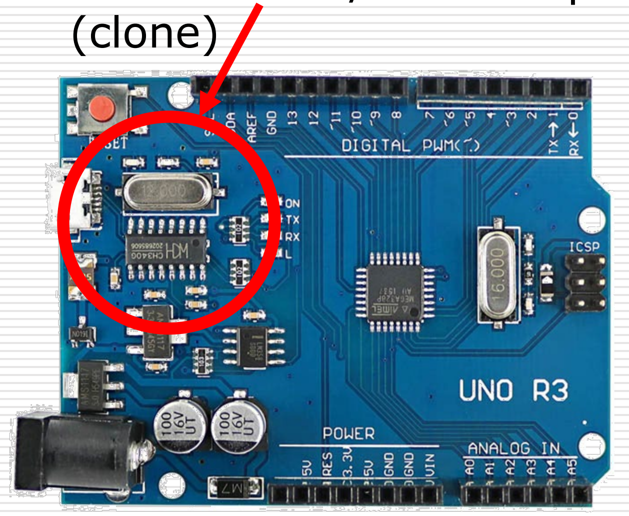
UNO board drivers

- Sometimes drivers to be installed.
  - ❑ Original Arduino boards use the FTDI drivers,
  - ❑ OEM boards use the cheaper CH340 drivers, which need to be installed.
- Plenty of help using Google

FTDI USB/Serial chip (original)



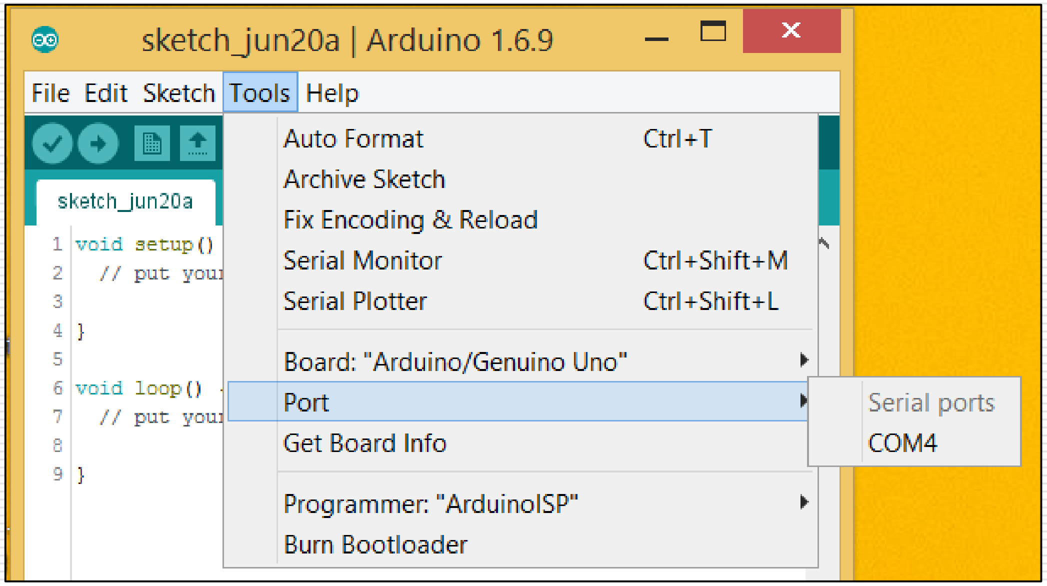
CH340G USB/Serial chip (clone)



Ref: [How to install Cheap China Arduinos that come with theCH340G/341G Serial/USB chip]()

Using the Arduino IDE

- You need to connect your UNO board to the host computer.
- Launch the Arduino IDE
- Setup the IDE
- Select the correct board that you are using (Tools>Board)



```
void setup() {
  // put your
}

void loop() {
  // put your
}
```

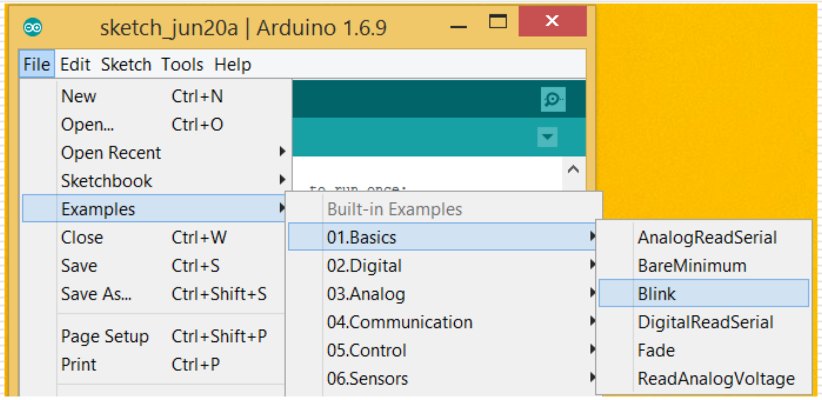
Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Serial Monitor	Ctrl+Shift+M
Serial Plotter	Ctrl+Shift+L
Board: "Arduino/Genuino Uno"	
Port	
Get Board Info	
Programmer: "ArduinoISP"	
Burn Bootloader	

Serial ports
COM4

- Identify and check the port the board is connected to (Tools > Serial Port > (select the COM port))

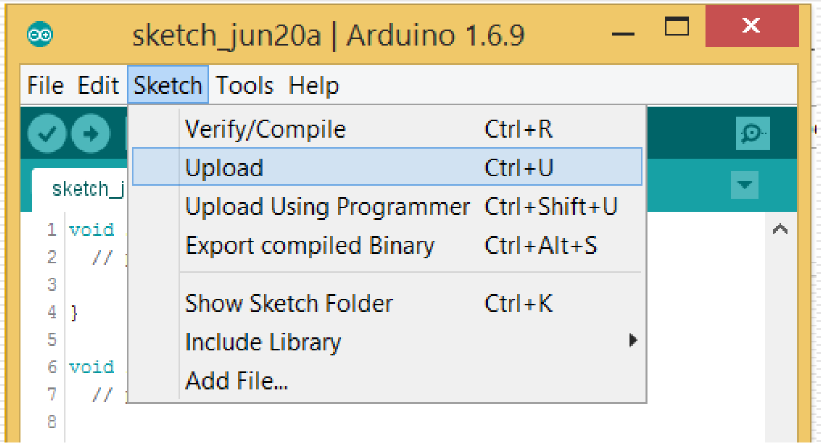
Test a Sample program

- Load the example program “Blink”.
- Programs are called Sketches.



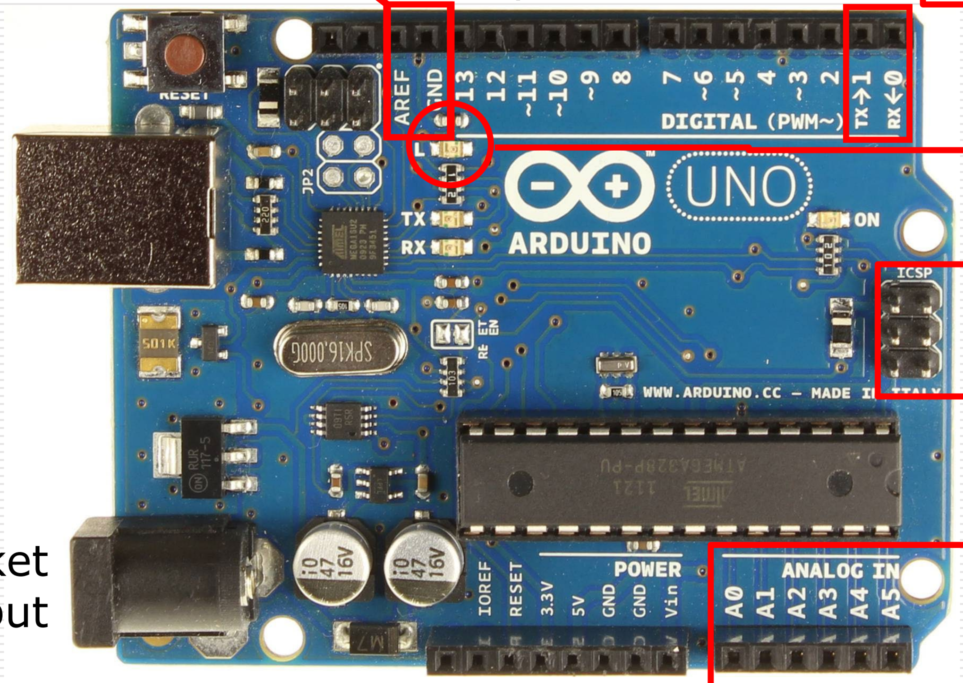
- Verify/Compile the program
- Upload

- Program executes after loading

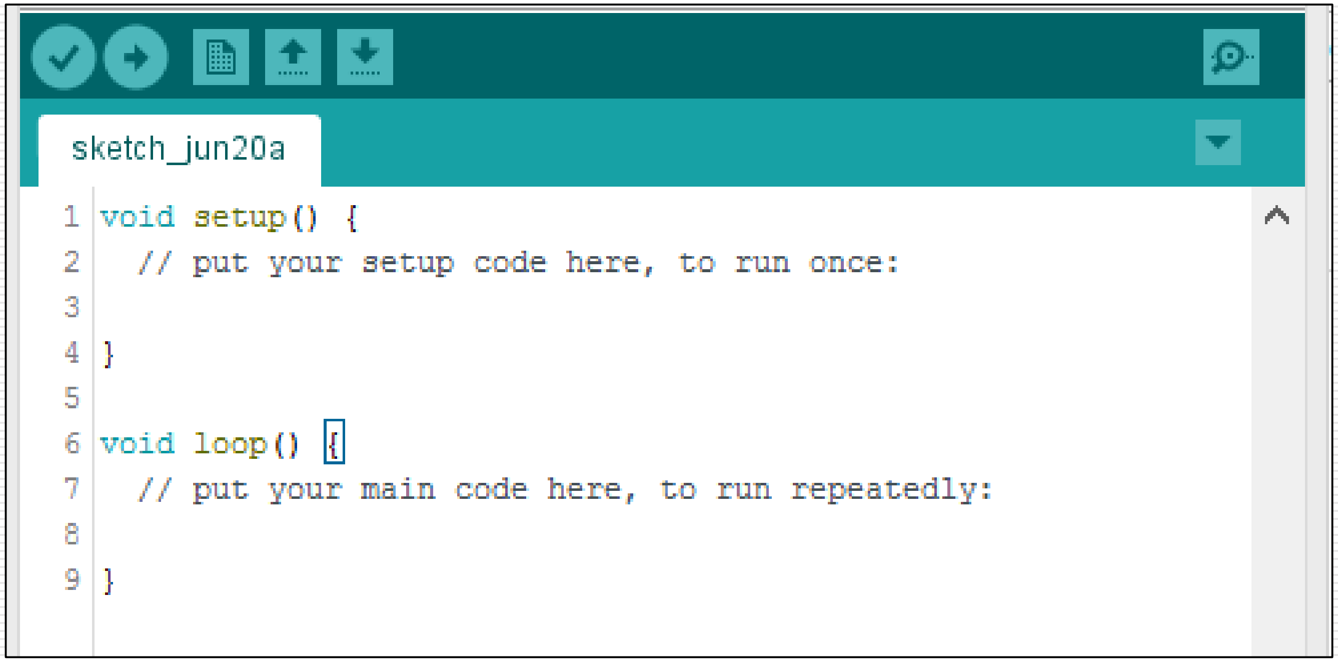

sketch_jun20a | Arduino 1.6.9
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...

Upload with programmer is only used with an ISP circuit

UNO board interfaces



The ARDUINO program



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

}

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

}
```

- Called a Sketch (extension .ino)
- Code in the setup function is executed at the start and only once.
- Code in the loop function is executed continually after setup() is run.

setup()

- Executed only ONCE after each powerup or reset of the UNO.
- UNO is automatically reset after each successful sketch upload
- Place
  - ❑ Initialization code here
  - ❑ Initialize your variables
  - ❑ Initialise your I/O pins here
- Tip: use identifiers to name your I/O pins, it makes programming much easier

loop()

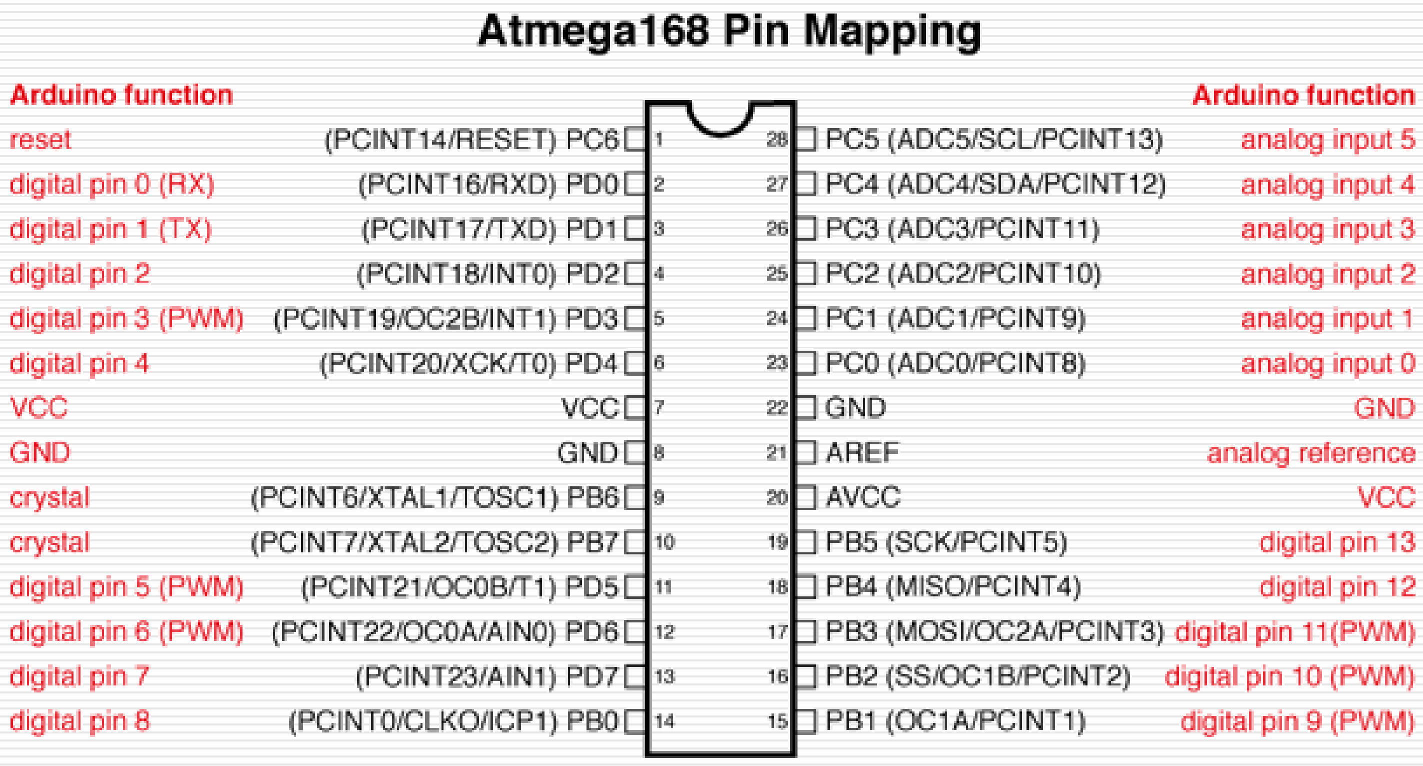
- After execution of the setup() function, the loop() function is executed.
- Loops infinitely, executing the code within the loop.
- Place your code/program within this function (there is no STOPping this code)

- Arduino code is based on C++.
- Follow good C++ programming habits:
  - Use comments (// or /* .. */)
  - Indent your code
  - Use UPPERcase to denote constants or defines

Digital Input/Output

- ATMega328 has 14 digital input/output ports.
- Digital values (1 = 5V, 0 = 0V)
- Some of these ports are multifunctional, depending on how they are initialised.
- They can perform as
  - ❑ Digital inputs (defaults)
  - ❑ Digital outputs
  - ❑ Pulse-width modulation outputs
- Arduino provides useful library functions for these purposes, simplifying programming.

Atmega328 – Arduino pin mapping



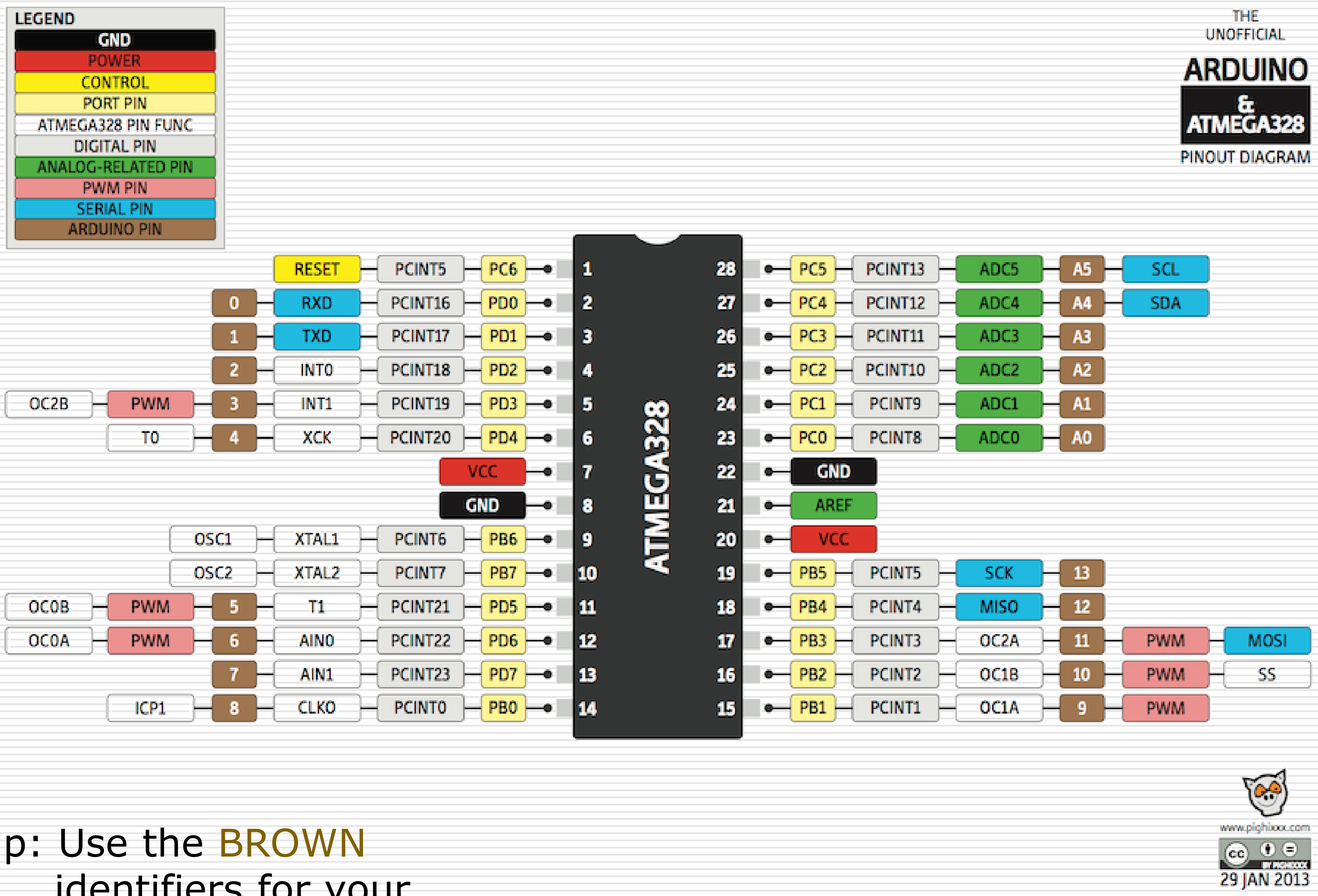
Atmega168 Pin Mapping

Arduino function		Pin	Pin		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 MOSI, MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Ref: Arduino – ATmega328 Pin Mapping

Atmega328 – Arduino pin mapping


LEGEND
GND
POWER
CONTROL
PORT PIN
ATMEGA328 PIN FUNC
DIGITAL PIN
ANALOG-RELATED PIN
PWM PIN
SERIAL PIN
ARDUINO PIN
THE UNOFFICIAL ARDUINO & ATMEGA328 PINOUT DIAGRAM
ATMEGA328
RESET — PCINT5 — PC6 — 1
0 — RXD — PCINT16 — PD0 — 2
1 — TXD — PCINT17 — PD1 — 3
2 — INT0 — PCINT18 — PD2 — 4
OC2B — PWM — 3 — INT1 — PCINT19 — PD3 — 5
T0 — 4 — XCK — PCINT20 — PD4 — 6
VCC — 7
GND — 8
OSC1 — XTAL1 — PCINT6 — PB6 — 9
OSC2 — XTAL2 — PCINT7 — PB7 — 10
OC0B — PWM — 5 — T1 — PCINT21 — PD5 — 11
OC0A — PWM — 6 — AIN0 — PCINT22 — PD6 — 12
7 — AIN1 — PCINT23 — PD7 — 13
ICP1 — 8 — CLKO — PCINT0 — PB0 — 14
28 — PC5 — PCINT13 — ADC5 — A5 — SCL
27 — PC4 — PCINT12 — ADC4 — A4 — SDA
26 — PC3 — PCINT11 — ADC3 — A3
25 — PC2 — PCINT10 — ADC2 — A2
24 — PC1 — PCINT9 — ADC1 — A1
23 — PC0 — PCINT8 — ADC0 — A0
22 — GND
21 — AREF
20 — VCC
19 — PB5 — PCINT5 — SCK — 13
18 — PB4 — PCINT4 — MISO — 12
17 — PB3 — PCINT3 — OC2A — 11 — PWM — MOSI
16 — PB2 — PCINT2 — OC1B — 10 — PWM — SS
15 — PB1 — PCINT1 — OC1A — 9 — PWM
www.pighixxx.com
29 JAN 2013

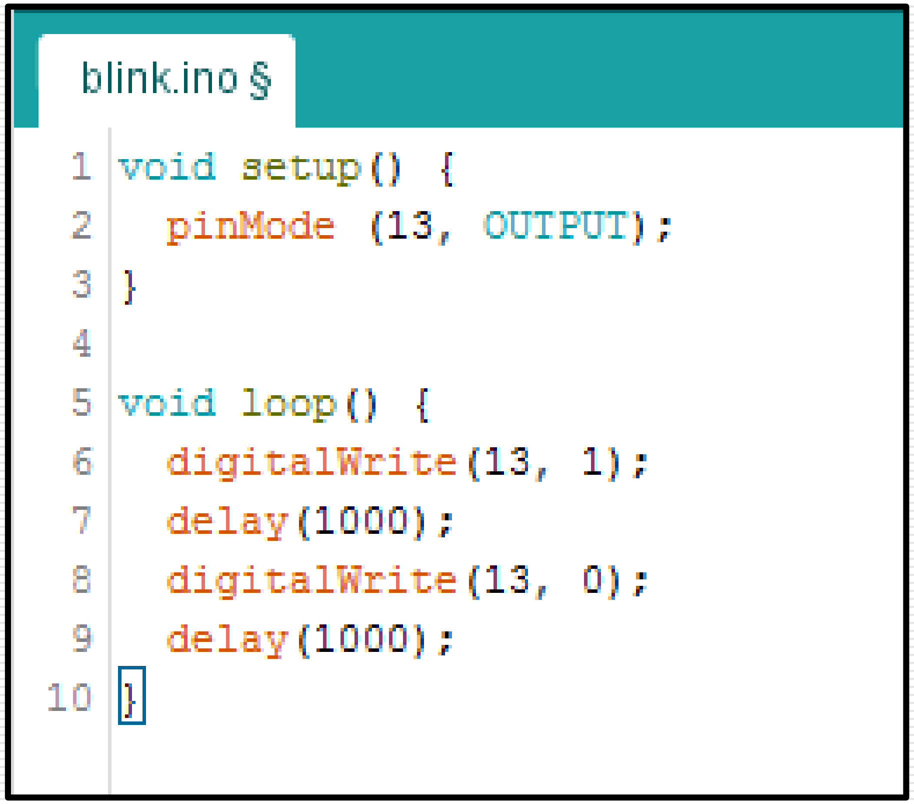
Tip: Use the BROWN identifiers for your Arduino sketch

Ref: [Pighxxx ATMega328 Pinout]

Digital Output

- pinMode()
  Initialise digital pin 13 to be a output port
- Repeat
  - ❑ digitalWrite()
    Turn ON the LED
  - ❑ delay()
    Wait 1 second
  - ❑ Turn OFF the LED
  - ❑ Wait 1 second

- [Arduino Programming reference]



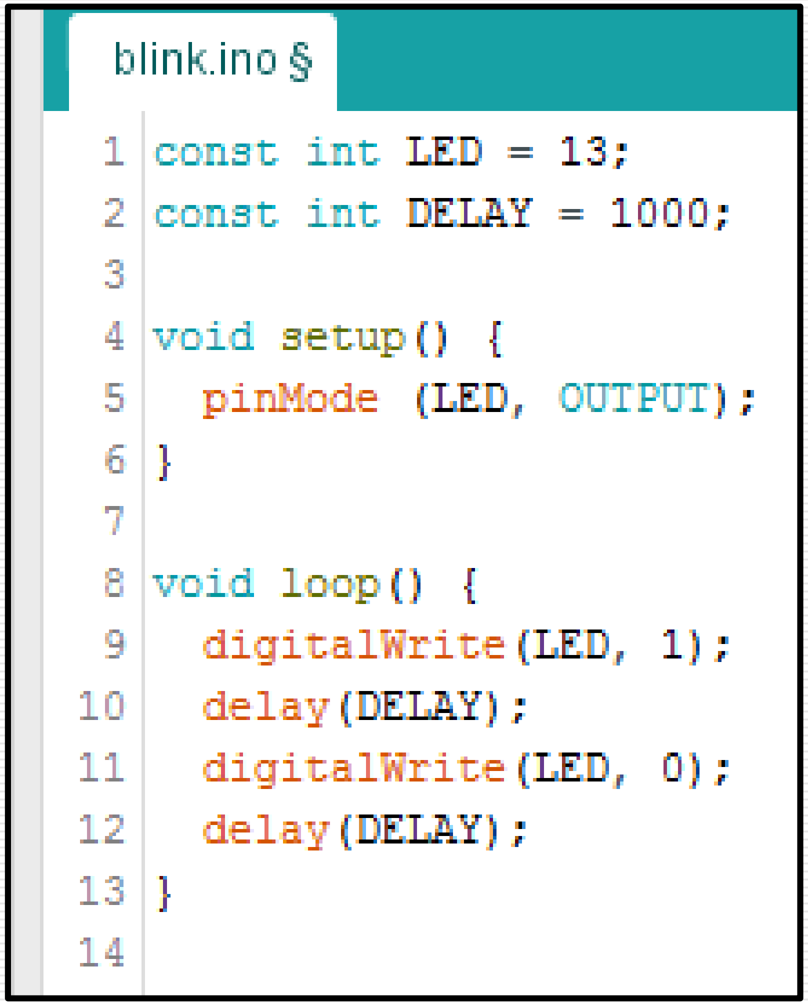
```
void setup() {
  pinMode (13, OUTPUT);
}

void loop() {
  digitalWrite(13, 1);
  delay(1000);
  digitalWrite(13, 0);
  delay(1000);
}
```

Colour coding helps in recognizing in-built functions, reserved words, values

Using identifiers

- Name the ports that you use, it makes it easier to change, configure, understand.
- Examine the following code.
  - ❑ How do I change the port from 13 to 4?
  - ❑ How do I change the delay to 0.5sec ?

UNO Port 13 is wired to a LED, useful for testing!



```
const int LED = 13;
const int DELAY = 1000;

void setup() {
  pinMode (LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, 1);
  delay(DELAY);
  digitalWrite(LED, 0);
  delay(DELAY);
}
```

Arduino for Beginners

- Youtube: Arduino – Tutorial 1 Arduino for beginners

Embedded Programming with Arduino

Rodney Dorville