

# **420-N23-LA Introduction to IoT**

## Arduino Platform Overview

---

# Arduino

An embedded  
processor platform  
with countless  
input/output  
possibilities.

# The Arduino Environment in general

---

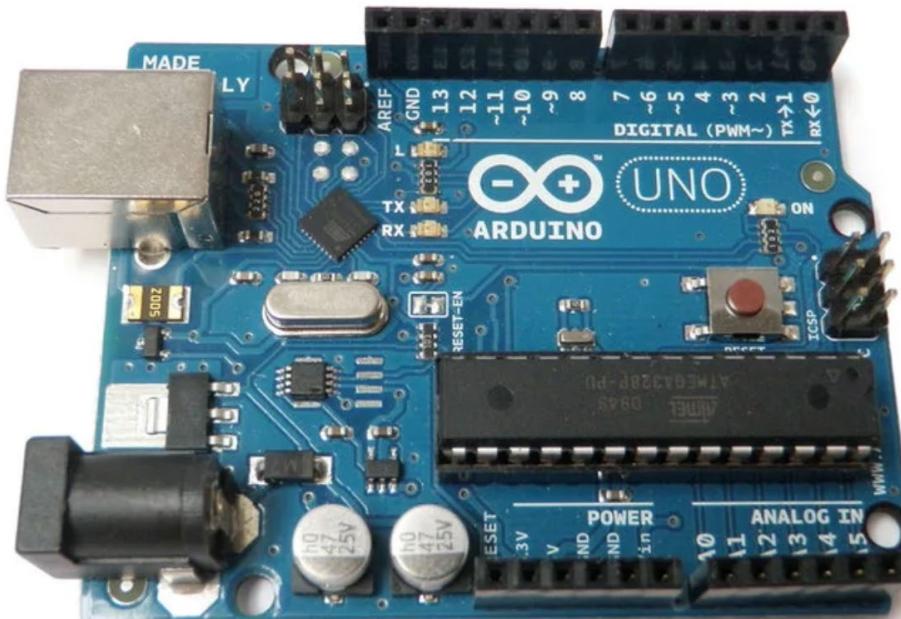
- **Arduino development board**
- **Shields**
- **Arduino IDE**

# What is an Arduino

---

- Arduino is an **open-source microcontroller development board**.
- It runs programs (instructions) that allow it to **interact with the physical world** through its **input and output ports**.

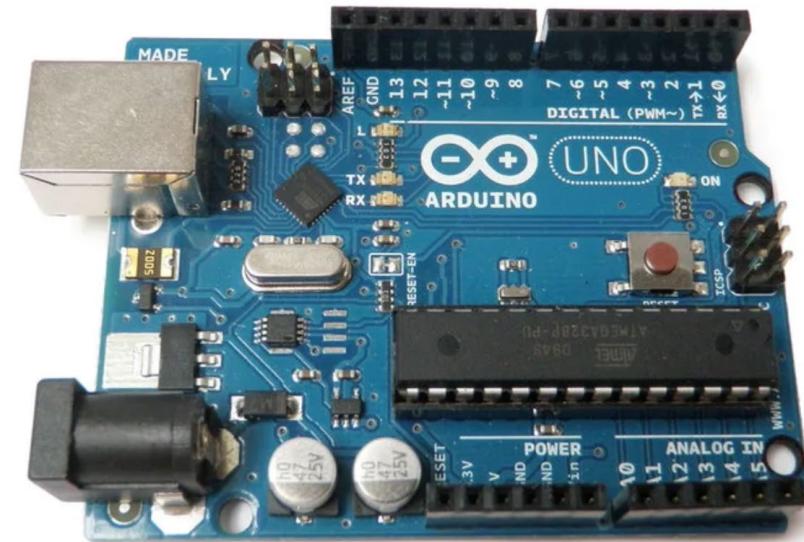
<https://www.elprocus.com/different-types-of-arduino-boards/>



# What is an Arduino

- With Arduino, you can:
  - **Read data** from sensors (e.g., temperature, light, motion).
  - **Control devices** such as motors, LEDs, and displays.
- Arduino is an example of an **embedded system**, a small computer designed to perform specific tasks.
- Think of Arduino as the “**brain**” that controls electronic devices!

<https://www.elprocus.com/different-types-of-arduino-boards/>



# What the Arduino is NOT

- ❑ Arduino is **not a full computer**
  - ❑ It does **not run a full operating system** like Windows, macOS, or Linux.
- ❑ Unlike a regular computer, Arduino boards (especially base models) **do not include**:
  - ❑ A real-time clock
  - ❑ Large memory (only very limited RAM & storage)
  - ❑ Built-in Wi-Fi or Bluetooth (unless specified on some models)
  - ❑ Audio support (no speaker or sound card)
  - ❑ Ethernet/network port
  - ❑ Built-in screen or display
  - ❑ Standard input devices (keyboard, mouse, touchpad)

👉 Arduino is a **microcontroller**, not a general-purpose computer.

This Arduino DOES have WIFI abilities built-in.  
(MKR1000).

<https://store.arduino.cc/usa/arduino-mkr1000-with-headers-mounted>



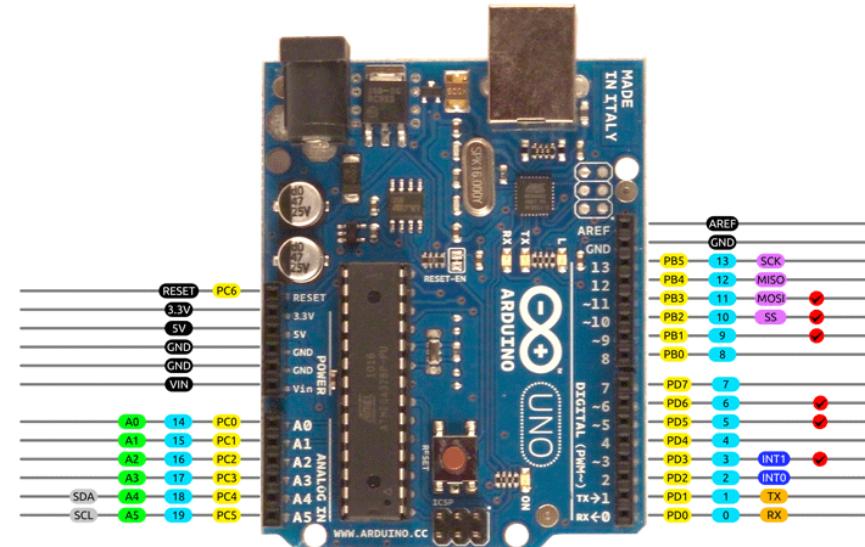
# Arduino Pins

- ❑ **Pins are connection points** on the board, used to:
  - ❑ **Receive input** (e.g., sensors, buttons)
  - ❑ **Send output** (e.g., LEDs, motors, displays)

- ❑ **Types of pins:**
  - ❑ **Digital Pins** → Read/write values as **HIGH (1)** or **LOW (0)**
  - ❑ **Analog Pins** → Read varying values (e.g., sensor data)

- ❑ **Power Pins:**
  - ❑ Provide **3.3V** or **5V** power to components
  - ❑ **GND (Ground)** connections for circuits

👉 Think of pins as the **bridge** between the Arduino and the outside world.



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

2014 by Bouni  
Photo by Arduino.cc

| Pin Category        | Pin Name   | Details   |
|---------------------|--|---|
| Power               | <b>Li-Po(3.7v),Vin, Vcc, 5V, GND</b>                   | <p><b>Lip-Po(3.7V):</b> The board can be powered by connecting a lithium polymer battery to this pin. The battery should have a nominal voltage of 3.7V and minimum of 700mAh</p> <p><b>Vin:</b> The board can also be powered by a regulated 5V supply connected to this input pin. The maximum voltage for this pin is 6V</p> <p><b>5V:</b> If powered through USB then this output pin can be used to get a +5V supply for powering other circuit</p> <p><b>Vcc:</b> This pin outputs a regulated 3.3V by using the on-board regulator IC.</p> <p><b>GND:</b> Ground pins.</p> |
| Reset               | <b>Reset</b>   | Resets the microcontroller.   |
| Analog Pins         | <b>A0 – A6</b>   | These 7 pins are used to measure analog voltage in the range of 0-3.3V, with a resolution of 8/10/12 bit  |
| DAC Pin             | <b>DAC0</b>  | Provides an analog voltage based in the digital input with a resolution of 10 bit   |
| Input/Output Pins   | <b>Digital Pins D0 - D14</b>                           | Can be used as input or output pins. 0V (low) and 3.3V (high)   |
| Serial              | <b>Rx, Tx</b>  | Used to receive and transmit TTL serial data.   |
| External Interrupts | 0, 1, 4, 5, 6, 7, 8, A1 -or 16-, A2 - or 17            | These 8 pins can be used as an external interrupt   |
| PWM                 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, A3 - or 18 -, A4 -or 19 | The 12 pins can be used to provide 8-bit PWM.   |
| SPI                 | 10(MOSI), 12 (MISO) and 9 (SCK)                        | Used for SPI communication.   |
| Inbuilt LED         | <b>13</b>  | To turn on the inbuilt LED.   |
| IIC                 | 11 (SDA), 12 (SCL)                                     | Used for I2C/TWI communication.   |
| AREF                | <b>AREF</b>  | To provide reference voltage for input voltage.   |



Description of what the Pins do (Will be studied in detail later in the course)

# Type of Pins

---

- ❑ **Connect the board to the outside world**

- ❑ Input → sensors, buttons
  - ❑ Output → LEDs, motors, displays

- ❑  **Types of Pins**

- ❑ **Digital** → ON/OFF (HIGH or LOW)

- ❑ **Analog** → Variable values (0–1023)

- ❑  **Power Pins**

- ❑ **3.3V / 5V** → supply power to components

- ❑ **GND** → ground connection

# Useful Power Pins

---

- ❑ **Vin** → Direct input voltage from adapter/USB
  - ⚠ Rarely used to power components directly
- ❑ **3.3V** → Regulated **3.3 volts** output for low-voltage modules
- ❑ **5V** → Regulated **5 volts** output for most components
- ❑ **GND (Ground)** → Common return path; every circuit must connect here

👉 These pins supply power to everything you connect to Arduino.

# Useful Communication Pins

## □ Rx / Tx → Serial Communication

- **TX** = Transmit data
- **RX** = Receive data

## □ SDA / SCL → I<sup>2</sup>C Communication

- **SCL** = Clock line (synchronizes data)
- **SDA** = Data line (carries info)

## □ AREF (Analog Reference)

- Allows using an **external voltage reference** for analog inputs
- Provides **greater accuracy** in measurements

👉 These pins are Arduino's “special tools” for data exchange and precision.

# Arduino Pins: Analog vs Digital

---

## 1 Digital Pins

- Can read **only two states: HIGH (1) or LOW (0)**
- Can **send signals** (OUTPUT) or **read signals** (INPUT)
- Examples: turning an LED ON/OFF, reading a button press

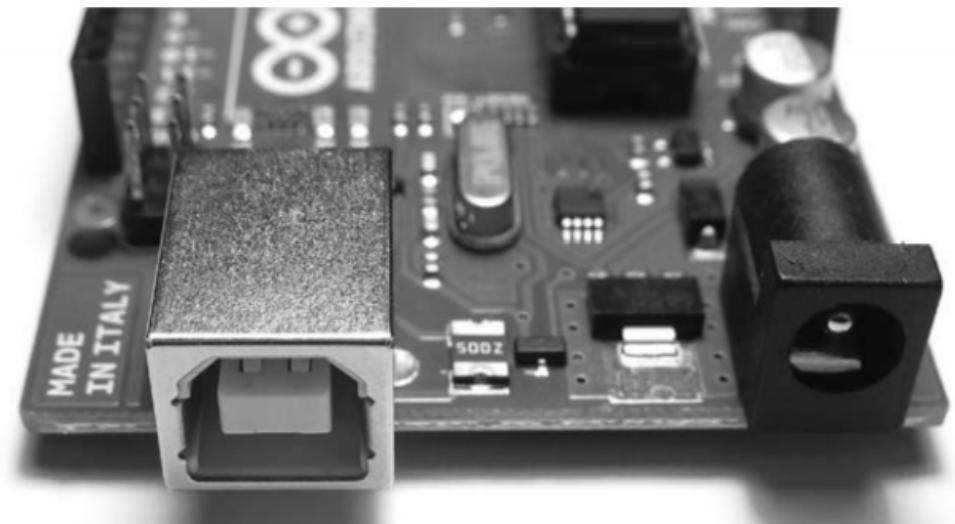
## 2 Analog Pins

- Can read **a range of values** (0–1023)
- Used for sensors that **give varying signals** like temperature or light sensors
- Example: reading a potentiometer to control LED brightness

# Arduino Board: USB and Power Connectors



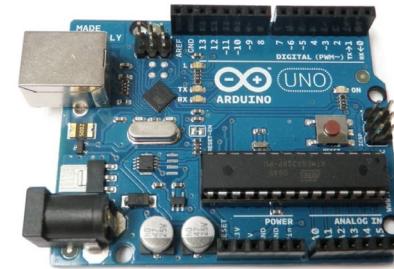
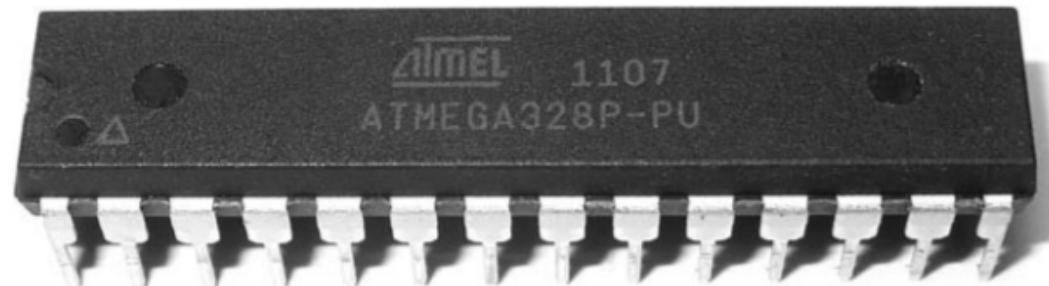
- To supply power to the board
- Upload your instructions to Arduino
- To send data and receive it from a computer



# Arduino Board: Microcontroller

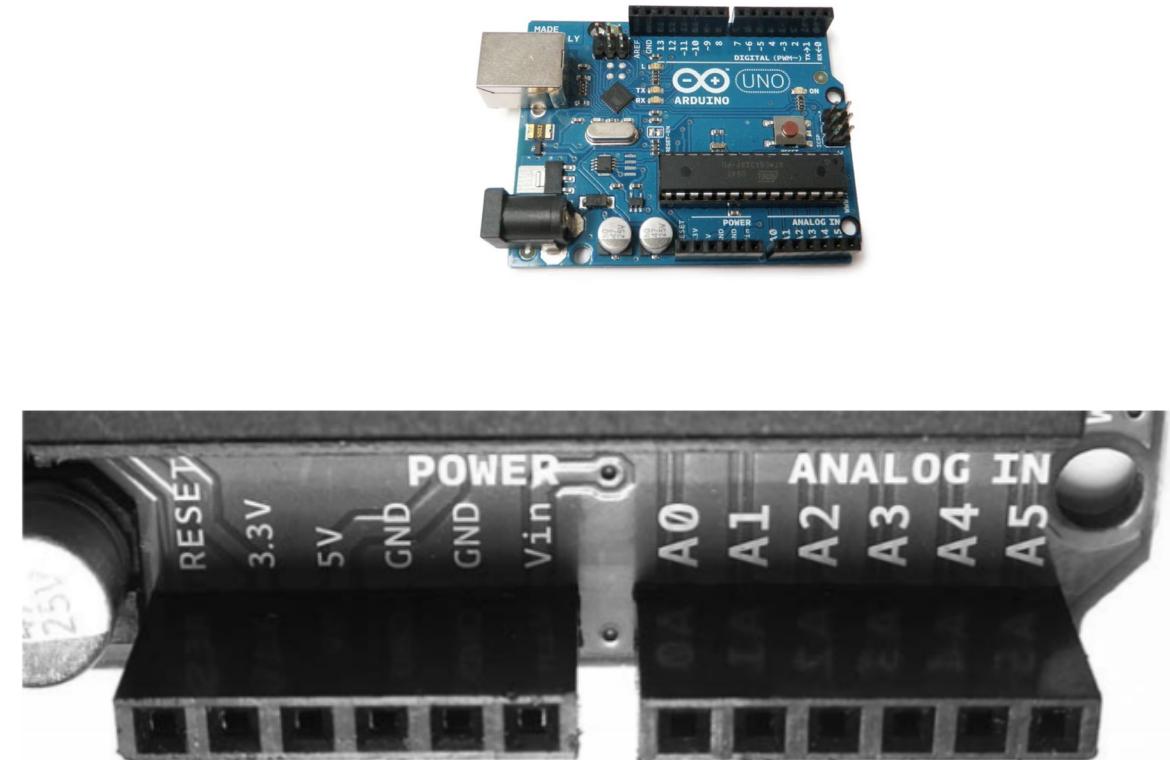
## What is Microcontroller?

- ❑ It is the brain of Arduino
- ❑ It is a tiny computer that contains:
  - ❑ a processor to execute instructions
  - ❑ Various types of memory to hold data and instructions
  - ❑ Provides various avenues of sending and receiving data.



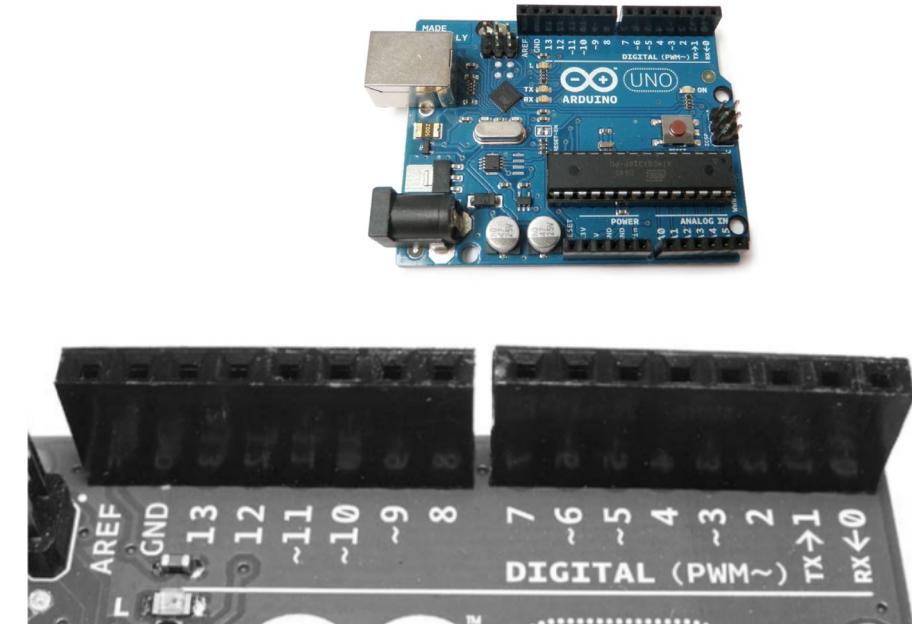
# Arduino Board: The Power and Analog Sockets

- First row offers:
  - Power connections
  - Ability to use an external RESET button.
  
- Second row offers:
  - Six analog input that are used to measure electrical signals that vary in voltage.
  - Pins A4 and A5 can also be used for sending data and receiving it from other devices.



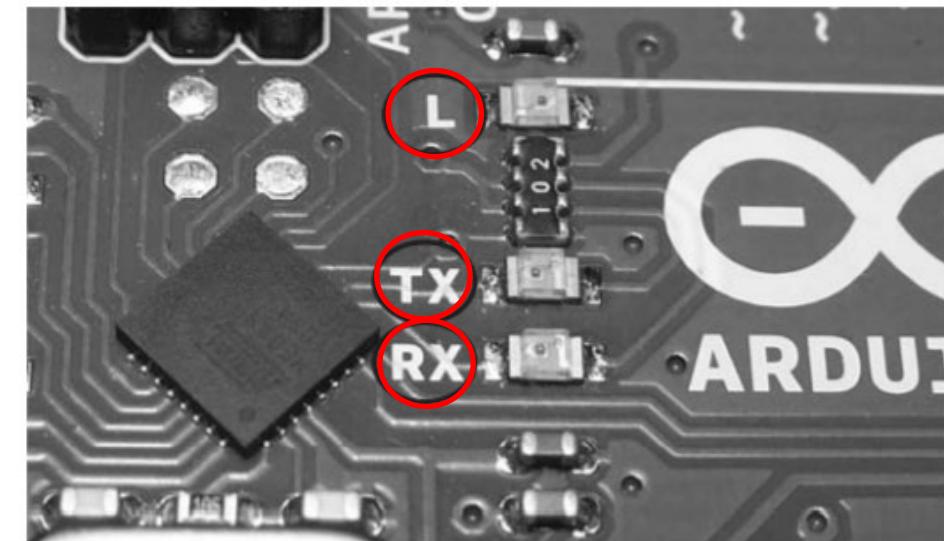
# Arduino Board: The Digital Input/Output Pins

- Numbered from 0 to 13.
- They can either detect
  - Whether or not an electrical signal is present OR
  - Generate a signal on command.
- Pins 0 and 1: serial port
  - used to send and receive data to other devices, such as a computer via the USB connector circuitry.
- The pins labeled with a tilde (~)
  - can also generate a varying electrical signal, which can be useful for such things as creating lighting effects or controlling electric motors.



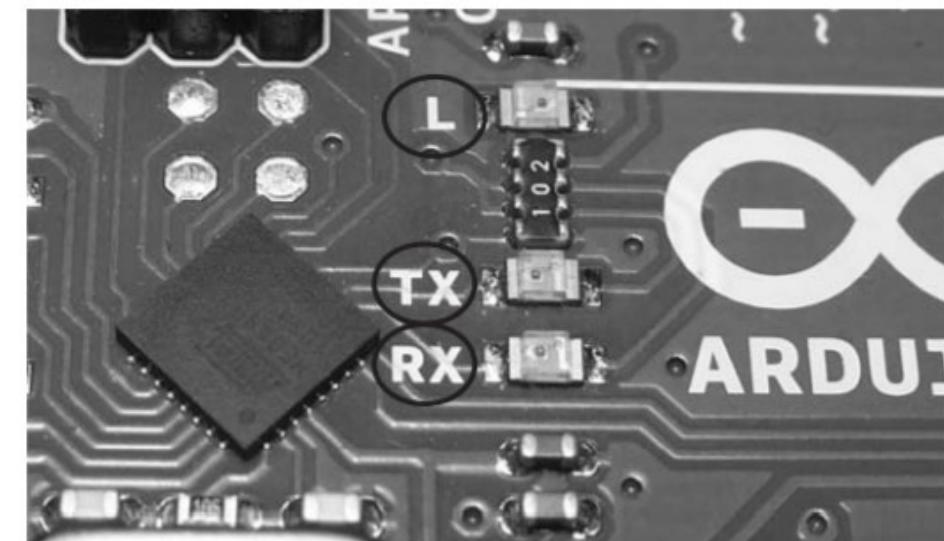
# Arduino Board: Light-Emitting Diodes (LEDs)

- The Arduino board has four LEDs:
  - one on the far right labeled **ON**
    - indicates when the board has power
  - Three in another group:
    - The LEDs labeled **TX** and **RX** light up when data is being transmitted or received between the Arduino and attached devices via the serial port and USB.
    - The **L LED** is for your own use (it is connected to the digital I/O pin number 13).



# Tiny Microcontroller to Control USB Interface

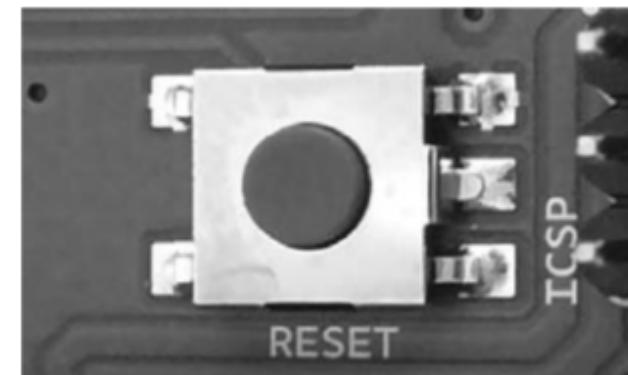
- The little black square part to the left of the LEDs is a tiny **microcontroller**
  - Controls the USB interface that allows your Arduino to send data to and receive it from a computer.



# Arduino Board: RESET Button

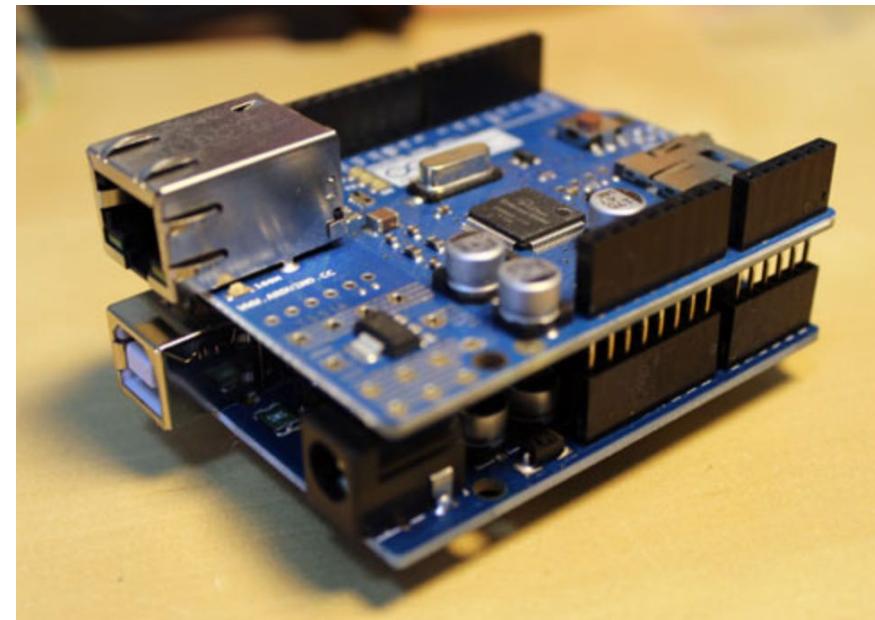
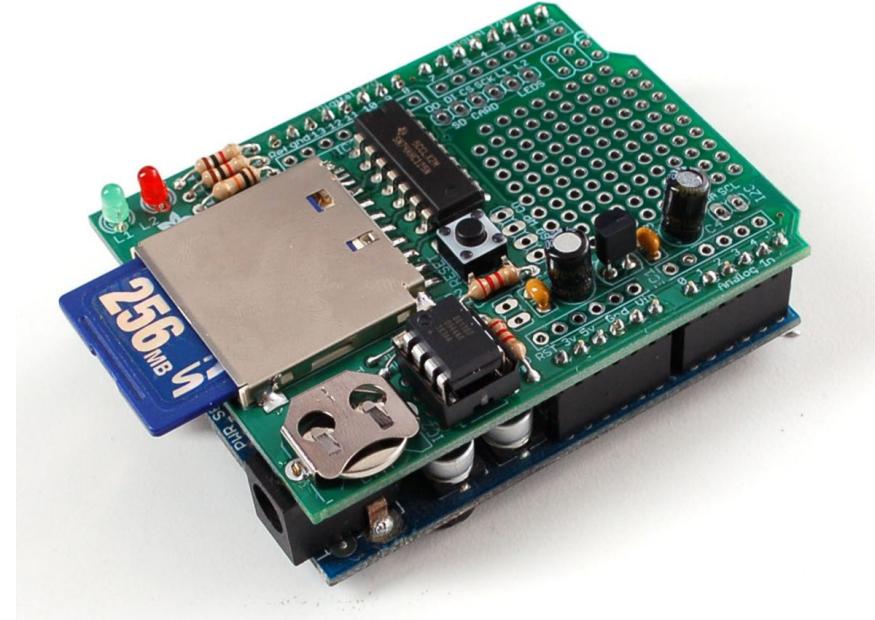
---

- This simple RESET button on the board is used to restart the system to resolve any problem.



# Shields

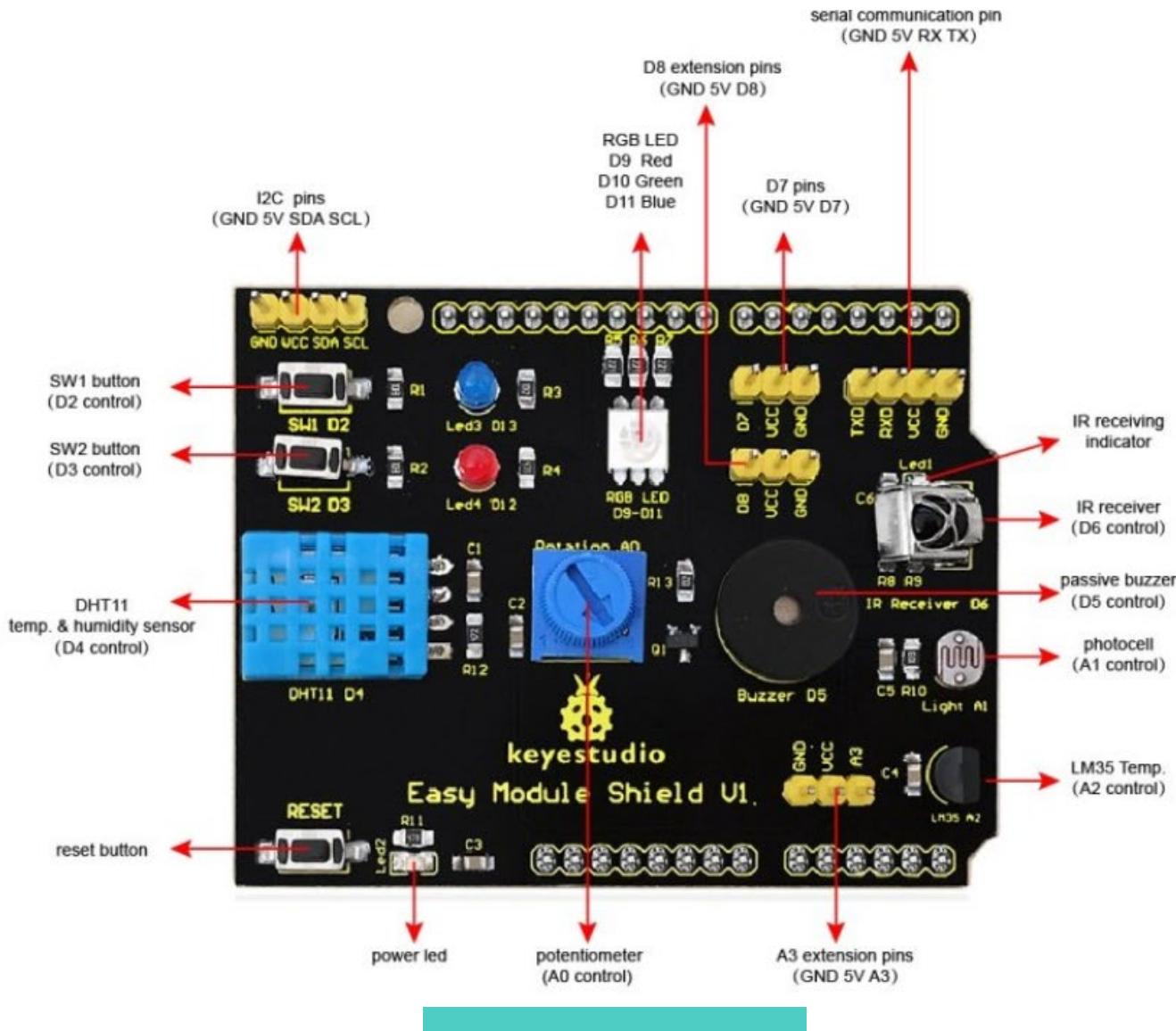
- Shields are expansion adapter boards that plug in over top of the Arduino Uno and gives it special functions.
- It might be faster to use a shield than to manually add parts (like a Wi-Fi chip) to the machine.
- Shields can be **stacked** (sometimes).
- *To the right, is a memory adapter module (top) and an ethernet network port (bottom).*



# Types of Shields

---

- ❑ Ethernet
  - ❑ Adds an Ethernet port, usually 100Mbps speed, not 1000.
- ❑ Relay
  - ❑ Control high-power devices (turn on and off ) with a relay.
- ❑ Protoshield
  - ❑ A breadboard on a shield.
- ❑ Motor Shield
  - ❑ Controls higher-power motors.
- ❑ LCD Shield
  - ❑ Adds a display to the Arduino.
- ❑ GPRS/GSM Shield, WIFI Shields, Bluetooth Shields
  - ❑ These shields are great to add WIFI to the machine easily.
  - ❑ GSM uses the cell network, this is good for mobile projects, or home-security projects.
- ❑ MicroSD Shield
  - ❑ Saves data to a card. Captured images or logging data.
- ❑ All-in-One Shields
  - ❑ Contain few things on it, like buzzer, temperature, buttons, etc.



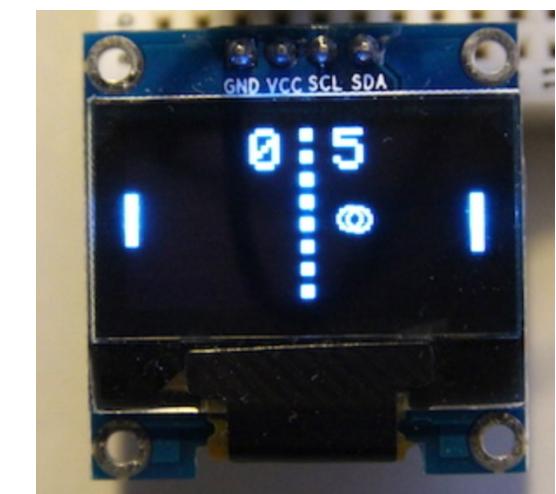
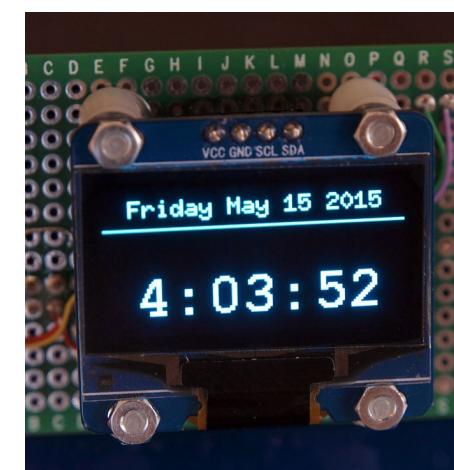
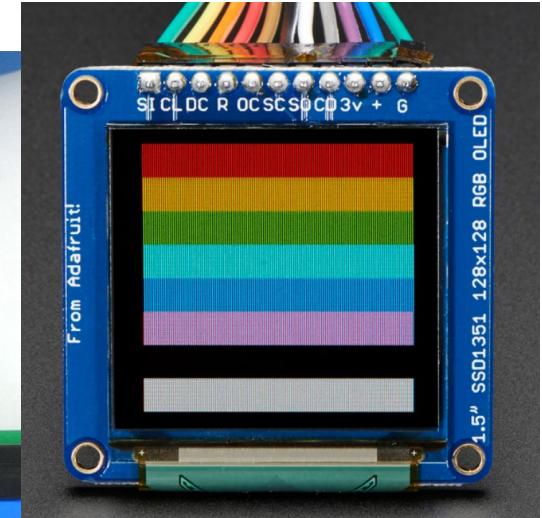
Easy Module Shield (All in One)

# Actuators

Output

# Displays

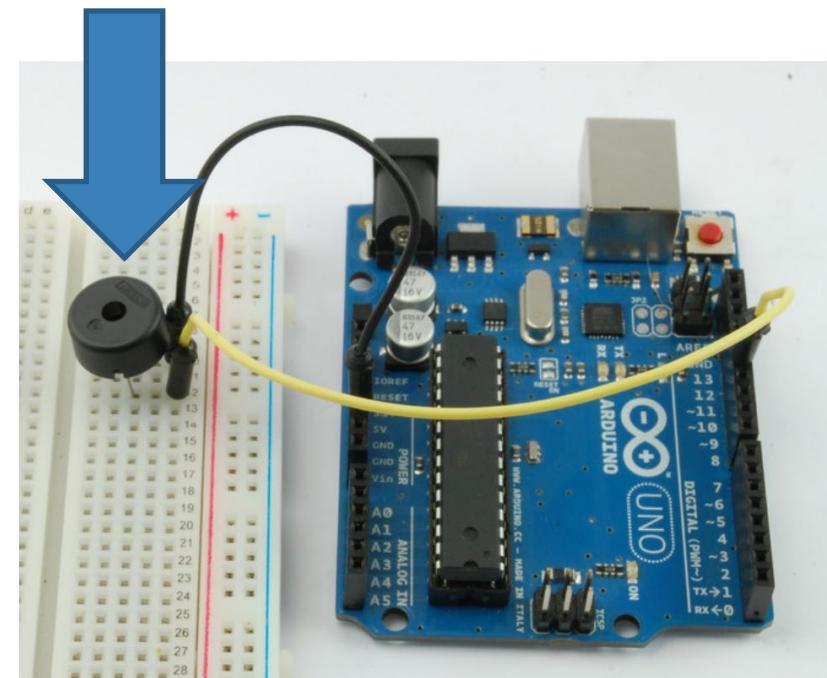
- Basic text and graphics available
- LCD
  - Has a constant backlight. A panel the same size as the **display** creates a steady white light that illuminates the **display**.
- OLED
  - Does not use a backlight. Each individual pixel produces its own light.
- Full Colour
  - <https://www.kickstarter.com/projects/rabidprototypes/the-pixelduino-an-arduino-with-an-awesome-oled-dis>



# Buzzers and Speakers

---

- Piezo buzzers require very little energy to work, they are perfect for Arduino.
- They don't sound good, but they do make basic alert sounds and beeps.



# Sensors

# Common Sensors

- HC-SR04 Ultrasonic Module



- Microphone Sensor



- RTC Clock



- HC-SR501 Pyroelectric Infrared Sensor



- DHT-11 Temperature Sensor



# Other Interesting Sensors

- These sensors are all very LOW COST.
- Generally you can find these on eBay or Amazon, overseas, under \$2 each. (Beware long shipping delays).
- <https://www.thegeekpub.com/wiki/list-of-arduino-sensors-and-modules/>

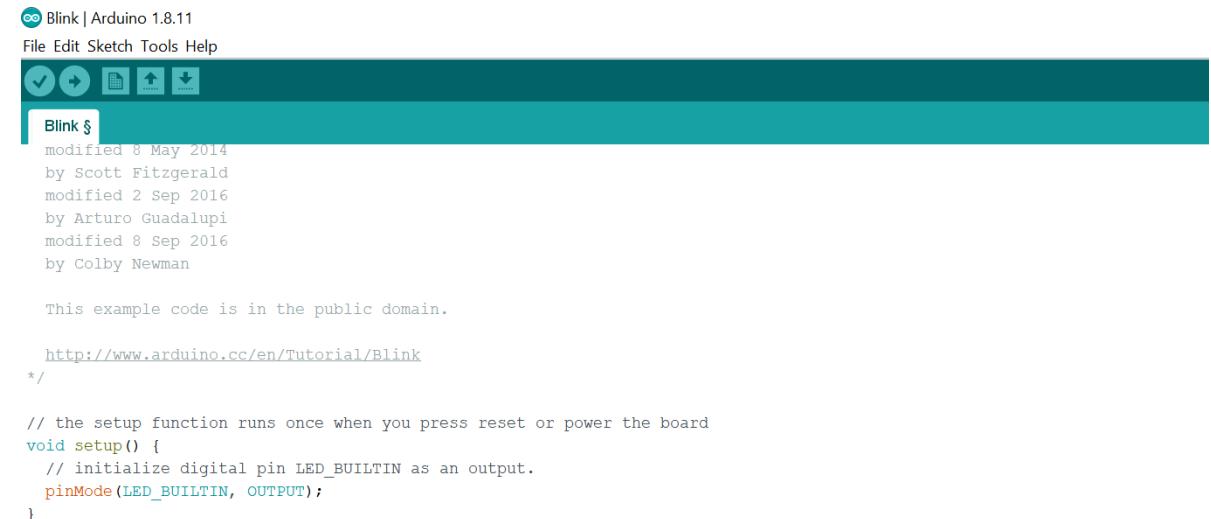
## RASPBERRY PI SENSORS AND MODULES

In most cases these sensors will also work with the [Raspberry Pi](#) without modification.

| Sensor/Module   | Description   | Links  |
|---|---|--|
|    | <b>Active Piezo-Buzzer Module</b><br>This active piezo-buzzer module KY-012 will make a beep sound (2.5 KHz) when 3.3V is applied to its input pins. No PWM required. You can see this in our <a href="#">Arduino Railroad Crossing project</a> . | <a href="#">Wiki Page</a><br><a href="#">Buy on Amazon</a> |
|    | <b>Flame IR Sensor</b><br>The KY-026 IR Flame sensor module is sensitive to the IR light spectrum emitted by open flames and triggers a digital output.   | <a href="#">Wiki Page</a><br><a href="#">Buy on Amazon</a> |
|   | <b>Heartbeat/Pulse Sensor</b><br>The KY-039 heartbeat detects heartbeat/pulse via the finger using a photoresistor and flashing an LED. Should <i>not</i> be used as a medical device.  | <a href="#">Wiki Page</a><br><a href="#">Buy on Amazon</a> |
|  | <b>High Sensitivity Microphone Sensor</b><br>This is the KY-037 sensor is a highly sensitive module for picking up sound and audio. This sensor triggers a digital output based on the intensity of the   | <a href="#">Wiki Page</a><br><a href="#">Buy on Amazon</a> |

# Arduino Software Environment (IDE)

- Cross compiler
- Debugger
- Programmer
- Simulator



The screenshot shows the Arduino IDE interface with the 'Blink' sketch open. The title bar reads 'Blink | Arduino 1.8.11'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu is a toolbar with icons for file operations. The code editor shows the 'Blink' sketch with the following content:

```
modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
 */

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}
```

- ❑ The Arduino IDE resembles a simple word processor.
- ❑ The IDE is divided into three main areas:
  - ❑ The command area
  - ❑ The text area
  - ❑ The message window area



File Edit Sketch Tools Help



sketch\_jun04a

#include <[SoftwareSerial.h](#)>

Software

bool D

Compile

Upload

(2,

for ESP8266

show more logs

< >

1

Compilation Logs –  
Other Logs

Arduino Uno on COM7

Arduino IDE

Serial Monitor Shortcut

Virtual COM port used,  
and Board Detected