Heaven's Light Is Our Guide

Rajshahi University of Engineering & Technology
Department of Computer Science & Engineering

CSE 3105
Computer Interfacing & Embedded System

Understanding the Arduino Landscape

Md. Nasif Osman Khansur
Lecturer
Dept. of CSE, RUET

References

1. Exploring Arduino 2e by Jeremy Blum [Chapter 1]
2. An Introduction to Arduino Uno Pinout [Link]

Introduction

What is Arduino?

The Arduino is a microcontroller development platform paired with an intuitive programming language that you develop using the Arduino integrated development environment based on easy-to-use hardware and software.

- The Arduino could be an automatic plant-watering control system.
- It could be a web server.
- It could even be a quadcopter autopilot and so on.

Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

- ✔ You can tell your board what to do by sending a set of instructions to the microcontroller on the board.
- ✔ To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming.

Introduction

Why Arduino?

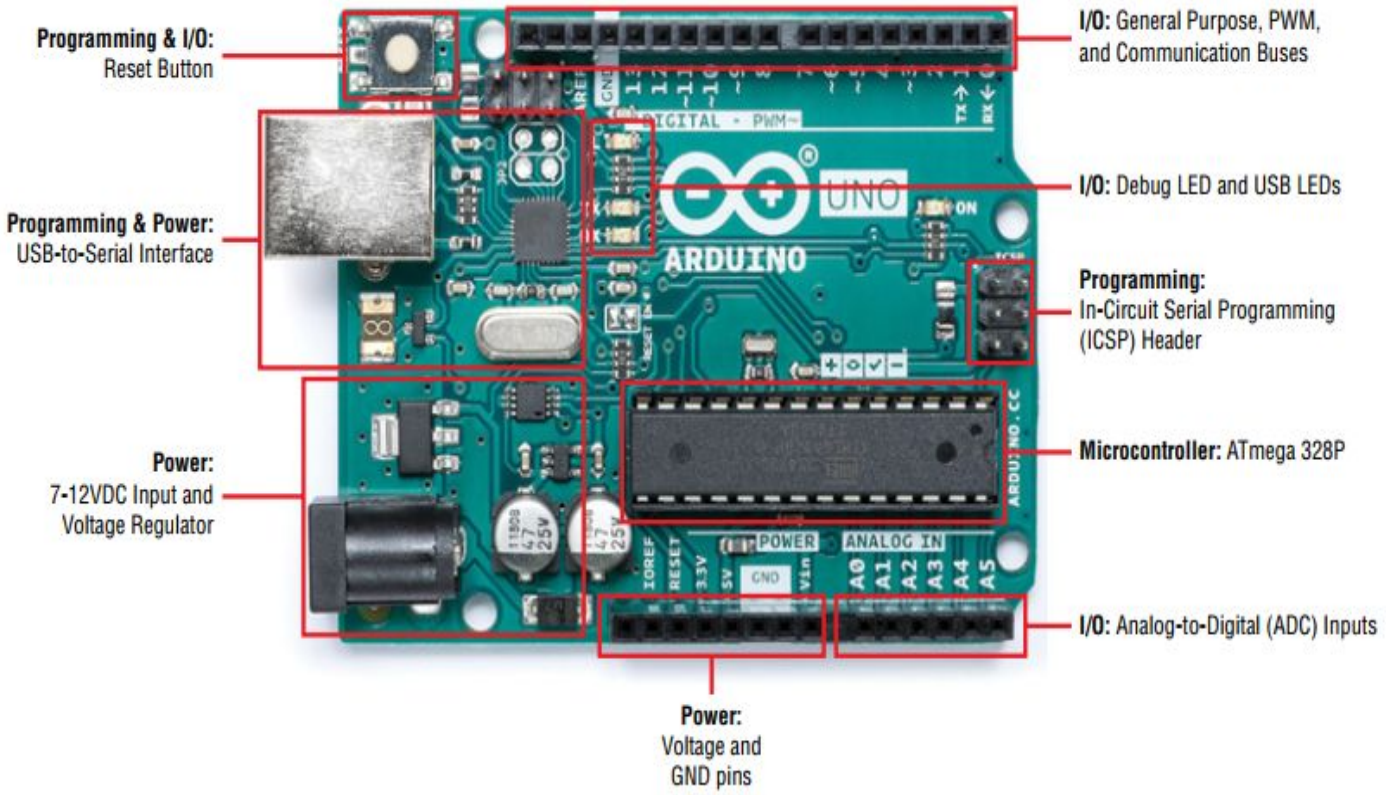
Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50

Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.

Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based.

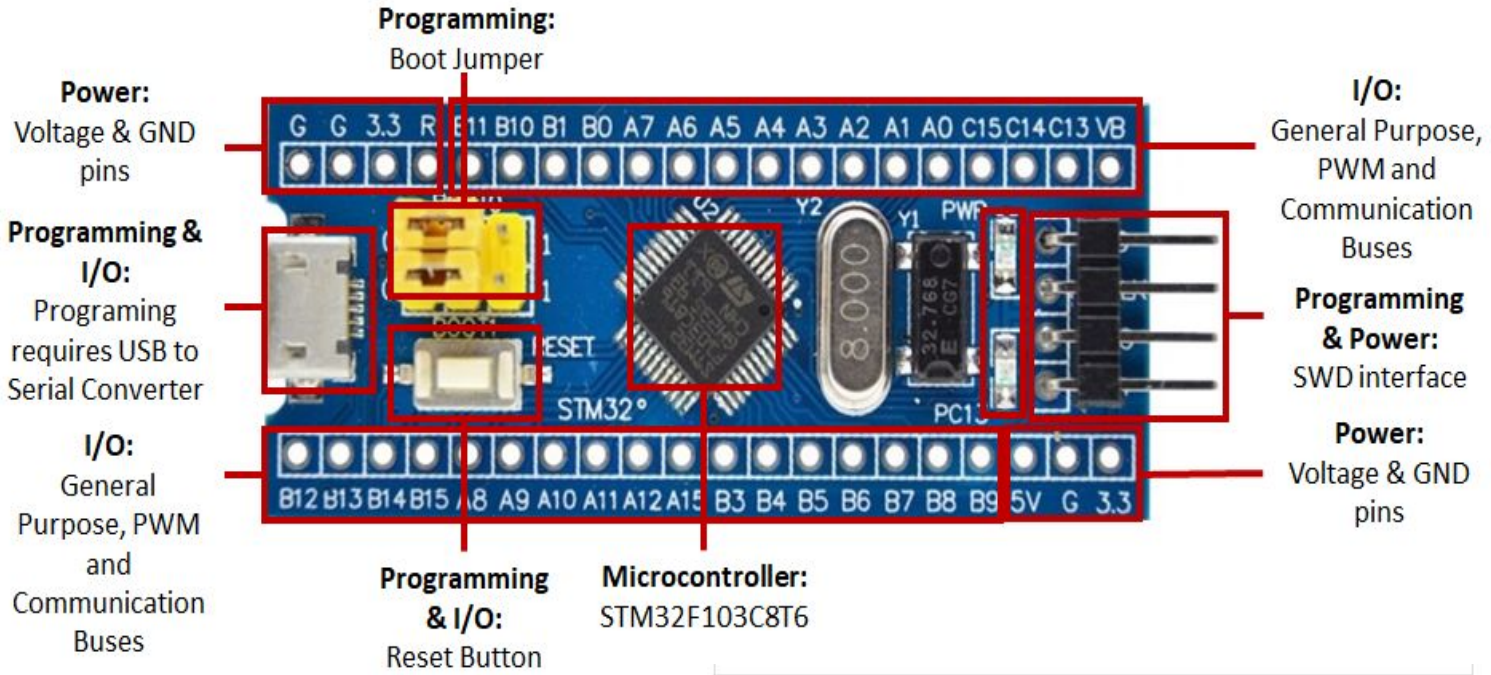
Arduino Uno Components



Functionality:

- *Microcontroller*: At the heart of every Arduino is a microcontroller. This is the brain of your Arduino.
- *Programming*: Programming interfaces enable you to load software onto your Arduino.
- *I/O*: Input/Output (I/O) circuitry is what enables your Arduino interface with sensors, actuators, etc.
- *Power*: There are a variety of ways to supply power to an Arduino. Most Arduino boards can automatically switch between power from multiple sources (such as USB and a battery).

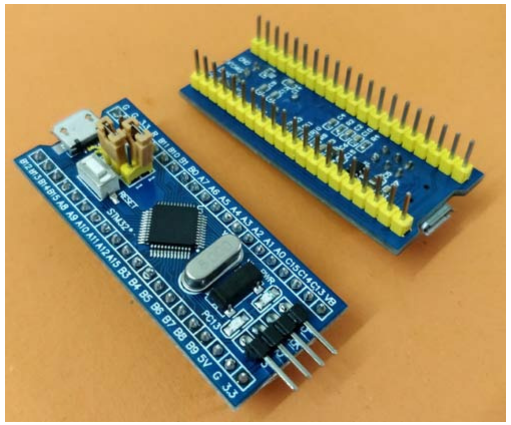
Blue Pill Components



Parameter	Meaning
STM	name of the manufacturer (STMicroelectronics)
32	32 bit ARM architecture
F	Foundation
1	Core (ARM Cortex M3)

03	Line (describes peripherals and speed)
C	48 pins
8	64 KB flash memory
T	LQFP package (Low Profile Quad Flat Pack)
6	Operating Temperature Range (-40 °C to 85 °C)

STM32 is a family of 32-bit microcontroller integrated circuits by STMicroelectronics. The STM32 chips are grouped into related series that are based around the same 32-bit ARM processor core.



The **STM32F103C8T6** (also known as 'STM32' or 'Blue Pill") is a cheap development board based on the ARM Cortex M3 microprocessor.

Arduino Uno vs. Blue Pill

Feature	Uno	STM32duino	Comments
MPU	ATMega328	STM32F103C8T6 (Cortex M3)	
Flash (KB)	32	64/128*	*128K in some models.
SRAM (KB)	2	20	
EEPROM (KB)	1	None	
Speed (MHz)		72	
Bus Width (bits)	8	32	
UARTs (HW)	1	3	Not all of the HW modules can be used at the same time due to pinout limitations. This applies to both types.
SPI	1	2	
I^2C (HW)	1	2	
Timers	2 (8 bit) + 1(16 bit)	3 (16 bits) + 1 (advanced type)	
Other	None	32.768KHz RTC crystal clock	
ADC (bits)	10	12	The Uno can use internal ADC reference, but the Blue Pill uses VCC reference. Thus, the Uno ADC can be more accurate.

Programming STM32

1) Using STM32duino bootloader (Arduino IDE)

You can program your STM32 development board using Arduino IDE, too. You will require FTDI (USB to UART converter) for this process.

2) Using Keil UVision and STM32CubeMX

This more professional in terms of usage. You will require the softwares ARM's Keil Uvision and STM32CubeMX for this method of programming BluePill. You will also need the STLink/V2 which is a debugger cum programmer hardware provided by STMicroelectronics. These softwares provide a more sophisticated and professional programming environment for programming embedded systems.

Microcontroller

A microcontroller (MCU) is a small, integrated computing device that contains a central processing unit (CPU), memory, input/output (I/O) peripherals, and other essential components on a single chip.

Microcontrollers are designed to execute specific tasks within embedded systems, such as controlling electronic devices, gathering sensor data, and processing information in real-time.

They are commonly used in a wide range of applications, including consumer electronics, industrial automation, automotive systems, medical devices, and IoT (Internet of Things) devices.

Microcontroller

Microcontroller Board: A microcontroller board, also known as a development board or evaluation board, is a hardware platform that incorporates a microcontroller chip along with additional components and interfaces to facilitate prototyping, experimentation, and development of embedded systems.

These boards typically provide features such as power regulation, clock circuitry, input/output pins, communication interfaces (e.g., USB, UART, SPI, I2C), and sometimes built-in sensors or LEDs.

Popular examples of microcontroller boards include the Arduino Uno, STM32 Blue Pill, and Raspberry Pi.

Microcontroller

Architecture: The architecture of a microcontroller refers to its internal design and organization of components, including the CPU (Central Processing Unit), memory, peripherals, and input/output interfaces.

Word Size: It represents the size of the binary data that the microcontroller's CPU can process or manipulate in a single instruction cycle. It also represents size of all addresses, integers, and other key data types.

For example, an 8-bit microcontroller architecture processes data in chunks of 8 bits at a time. Maximum size of addresses or data type is also 8 bit.

Manufacturer: The manufacturer of a microcontroller is the company that designs, produces, and sells the microcontroller.

Microcontroller

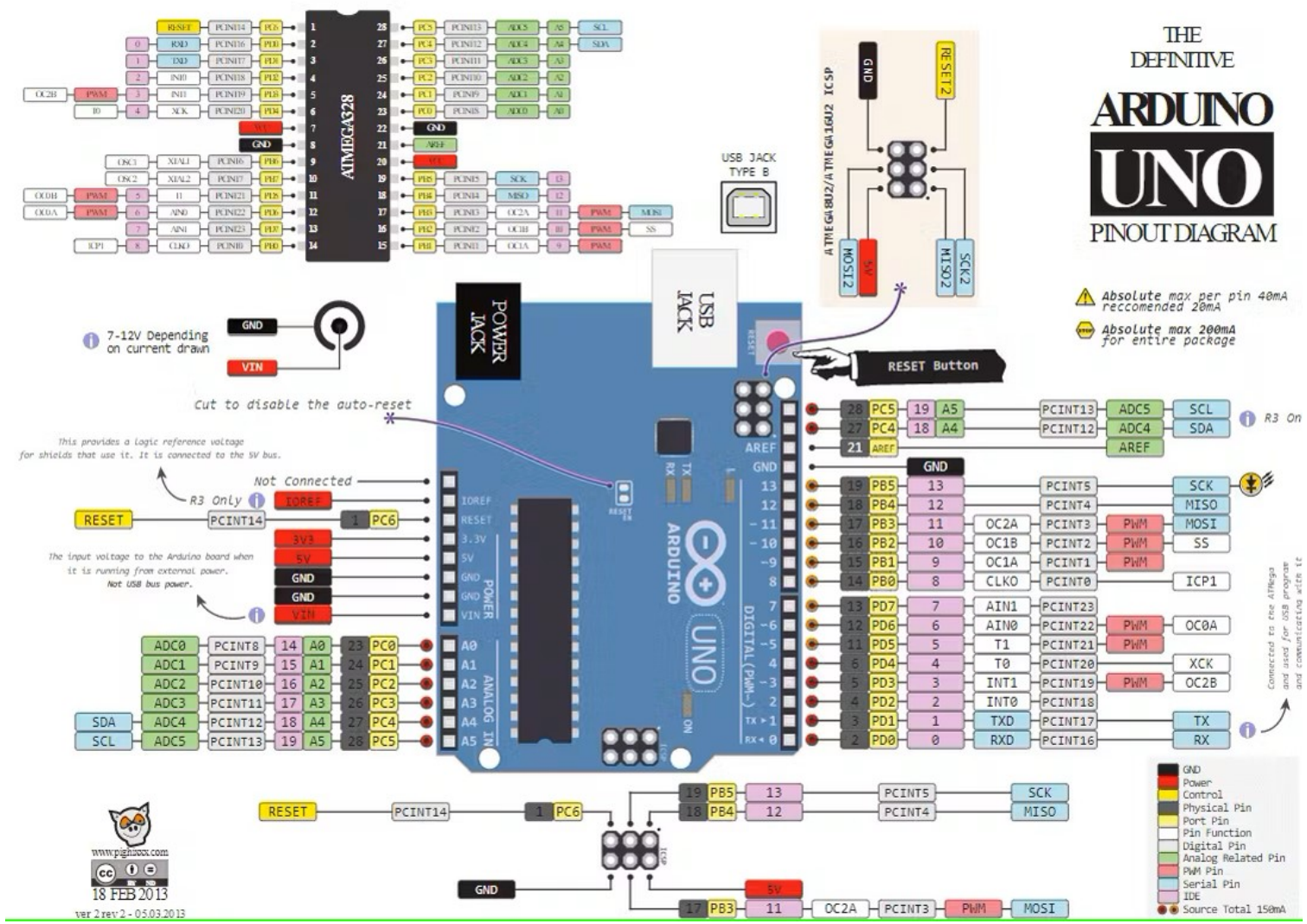
CPU Frequency: The CPU frequency, also known as clock speed, refers to the speed at which the microcontroller's central processing unit (CPU) executes instructions. It is measured in hertz (Hz) and determines the processing speed of the microcontroller.

Flash Memory: Flash memory is non-volatile memory used for storing program code (firmware) and data in a microcontroller. It retains its data even when power is removed from the microcontroller and is typically used to store the microcontroller's firmware.

RAM: RAM (Random Access Memory) is volatile memory used for temporary storage of data and variables during program execution.

Arduino Uno Diagram

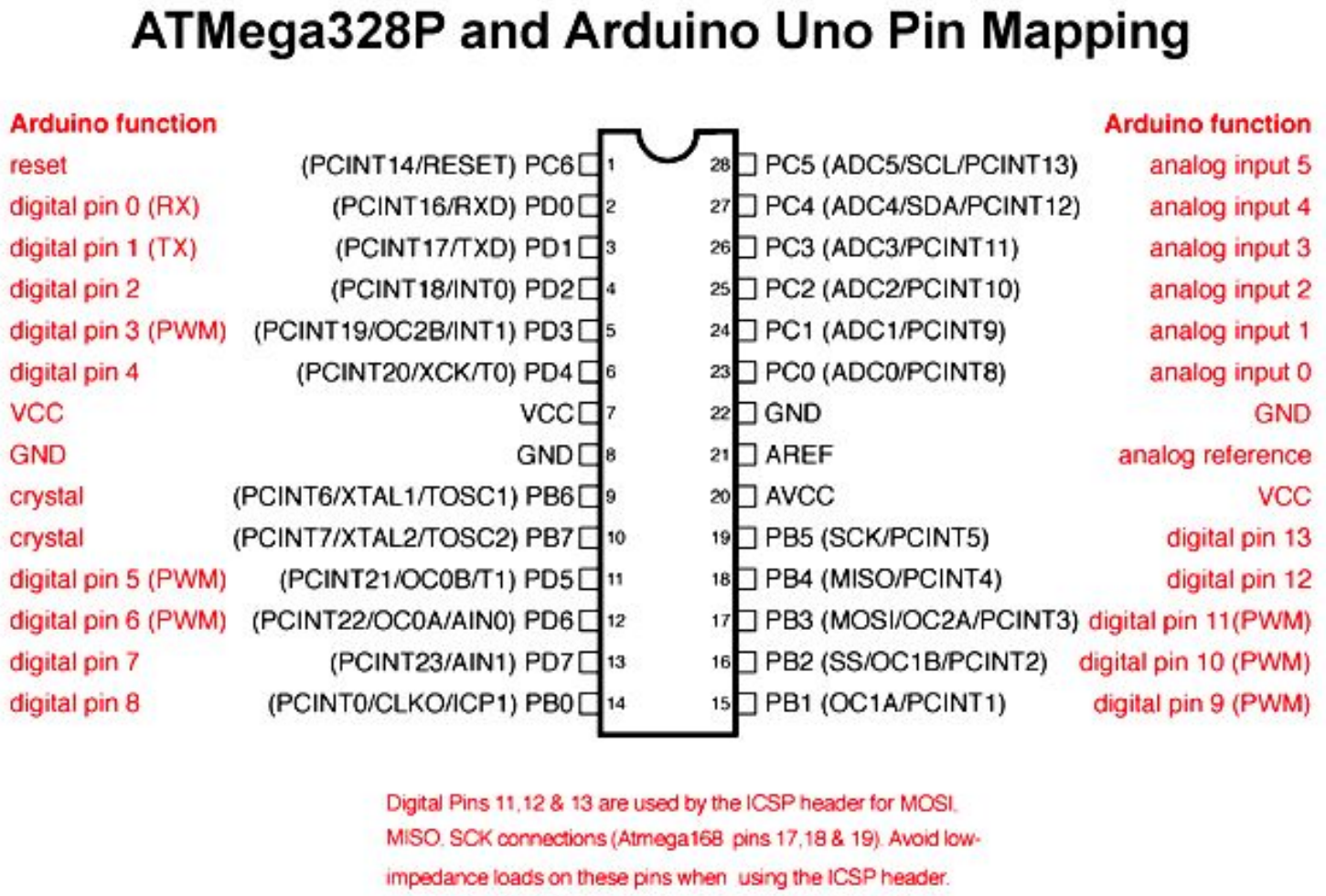
[Link]



ATMega328P

[Link]

328P =
‘32’ means 32 kByte of Flash or Program Memory (Code Space).
‘8’ means 8 bit architecture.
‘P’ stands for ‘picopower’; lower power consumption.



ATMega328P

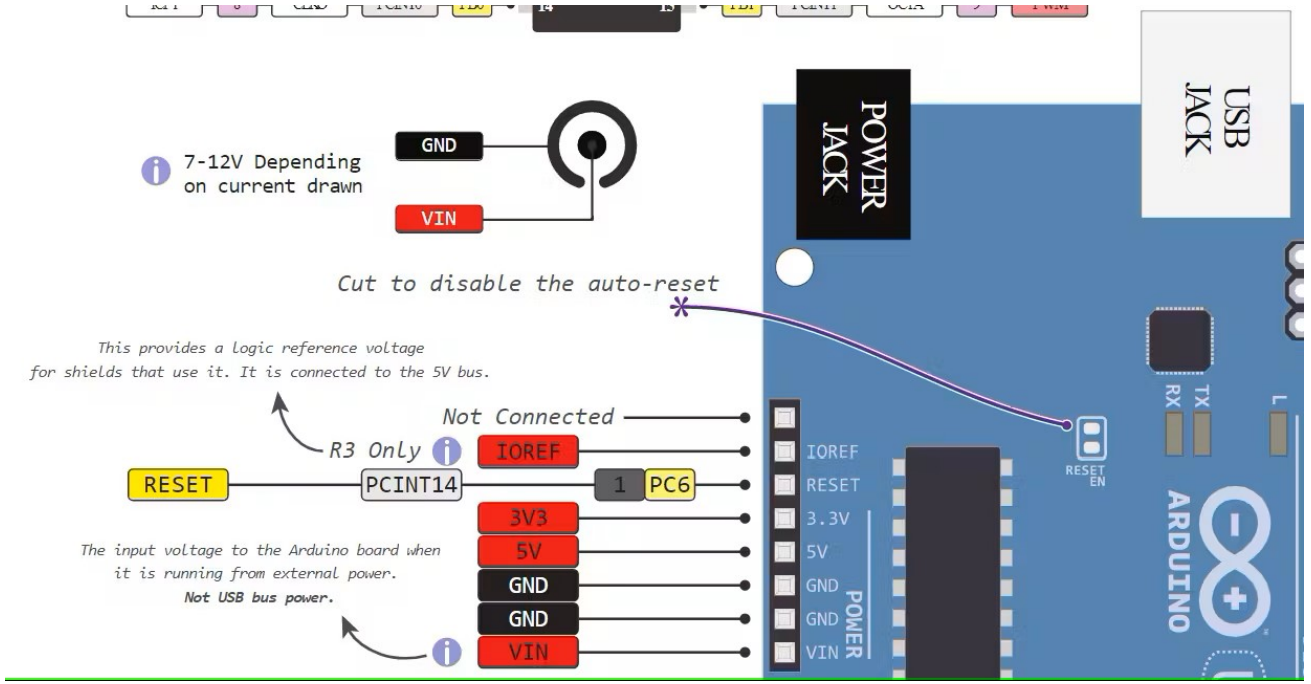
Pin No.	Pin name	Description	Secondary Function
1	PC6 (RESET)	Pin6 of PORT C	Pin by default is used as RESET pin. PC6 can only be used as I/O pin when RSTDISBL Fuse is programmed.
2	PD0 (RXD)	Pin0 of PORT D	RXD (Data Input Pin for USART) USART Serial Communication Interface [Can be used for programming]
3	PD1 (TXD)	Pin1 of PORT D	TXD (Data Output Pin for USART) USART Serial Communication Interface [Can be used for programming] INT2(External Interrupt 2 Input)
4	PD2 (INT0)	Pin2 of PORT D	External Interrupt source 0
5	PD3 (INT1/OC2B)	Pin3 of PORT D	External Interrupt source1 OC2B(PWM - Timer/Counter2 Output Compare Match B Output)
6	PD4 (XCK/T0)	Pin4 of PORT D	T0(Timer0 External Counter Input) XCK (USART External Clock I/O)
7	VCC		Connected to positive voltage
8	GND		Connected to ground
9	PB6 (XTAL1/TOSC1)	Pin6 of PORT B	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
10	PB7 (XTAL2/TOSC2)	Pin7 of PORTB	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
11	PD5 (T1/OC0B)	Pin5 of PORT D	T1(Timer1 External Counter Input) OC0B(PWM - Timer/Counter0 Output Compare Match B Output)
12	PD6 (AIN0/OC0A)	Pin6 of PORTD	AIN0(Analog Comparator Positive I/P) OC0A(PWM - Timer/Counter0 Output Compare Match A Output)
13	PD7 (AIN1)	Pin7 of PORTD	AIN1(Analog Comparator Negative I/P)
14	PB0 (ICP1/CLKO)	Pin0 of PORTB	ICP1(Timer/Counter1 Input Capture Pin) CLKO (Divided System Clock. The divided system clock can be output on the PB0 pin)
15	PB1 (OC1A)	Pin1 of PORTB	OC1A (Timer/Counter1 Output Compare Match A Output)
16	PB2 (SS/OC1B)	Pin2 of PORTB	SS (SPI Slave Select Input). This pin is low when controller acts as slave. [Serial Peripheral Interface (SPI) for programming] OC1B (Timer/Counter1 Output Compare Match B Output)
17	PB3 (MOSI/OC2A)	Pin3 of PORTB	MOSI (Master Output Slave Input). When controller acts as slave, the data is received by this pin. [Serial Peripheral Interface (SPI) for programming] OC2 (Timer/Counter2 Output Compare Match

ATMega328P

18	PB4 (MISO)	Pin4 of PORTB	MISO (Master Input Slave Output). When controller acts as slave, the data is sent to master by this controller through this pin. [Serial Peripheral Interface (SPI) for programming]
19	PB5 (SCK)	Pin5 of PORTB	SCK (SPI Bus Serial Clock). This is the clock shared between this controller and other system for accurate data transfer. [Serial Peripheral Interface (SPI) for programming]
20	AVCC		Power for Internal ADC Converter
21	AREF		Analog Reference Pin for ADC
22	GND		GROUND
23	PC0 (ADC0)	Pin0 of PORTC	ADC0 (ADC Input Channel 0)
24	PC1 (ADC1)	Pin1 of PORTC	ADC1 (ADC Input Channel 1)
25	PC2 (ADC2)	Pin2 of PORTC	ADC2 (ADC Input Channel 2)
26	PC3 (ADC3)	Pin3 of PORTC	ADC3 (ADC Input Channel 3)
27	PC4 (ADC4/SDA)	Pin4 of PORTC	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/output Line)
28	PC5 (ADC5/SCL)	Pin5 of PORTC	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)

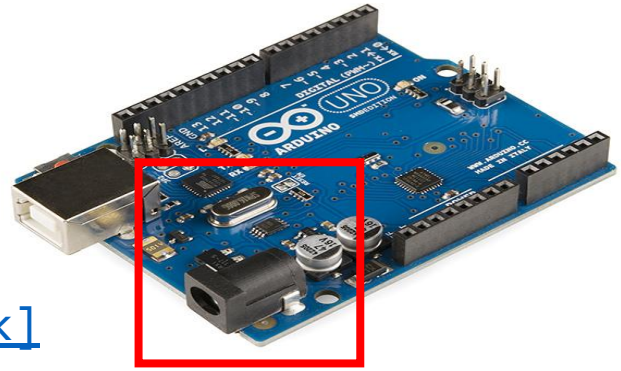
Power Supply

1. Barrel Jack
2. VIN Pin
3. USB cable
4. 5v and 3v3
5. GND
6. RESET
7. IOREF



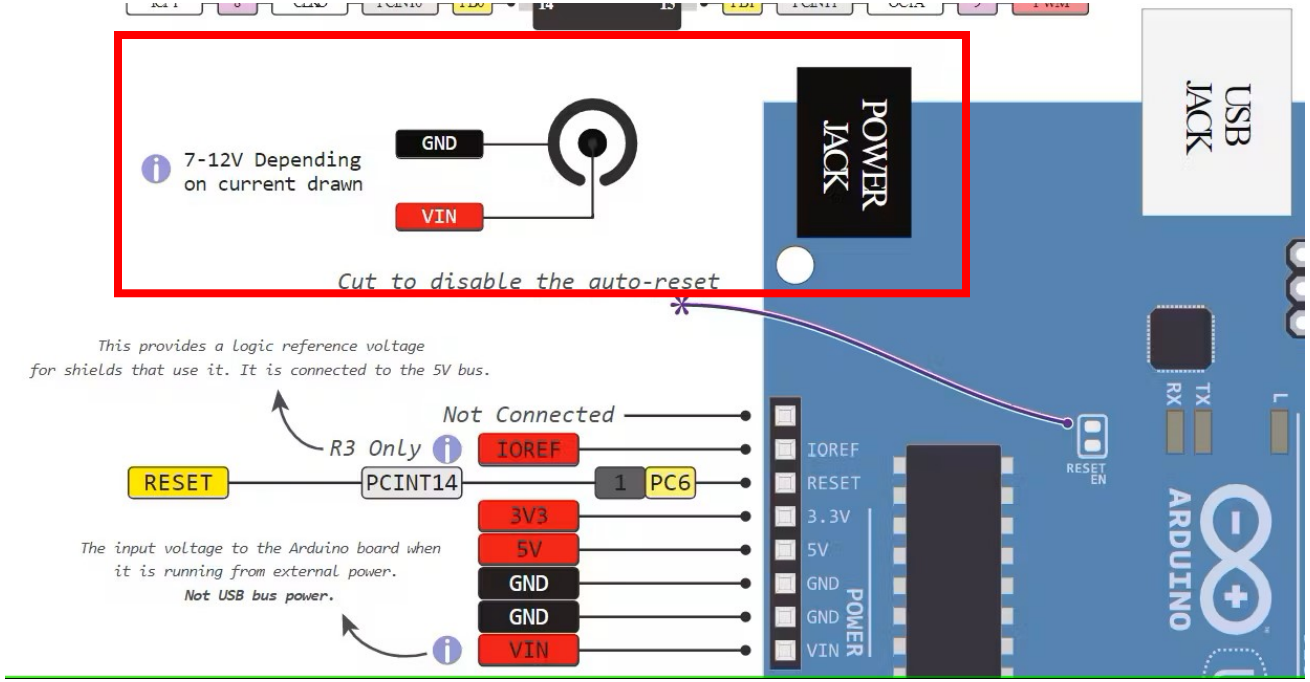
Power Supply

1. Barrel Jack
2. VIN Pin
3. USB cable



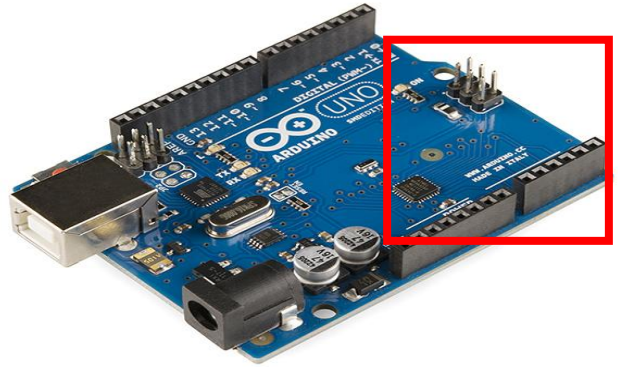
[Link]

The Barrel jack, or DC Power Jack can be used to power your Arduino board. The barrel jack is usually connected to a wall adapter.



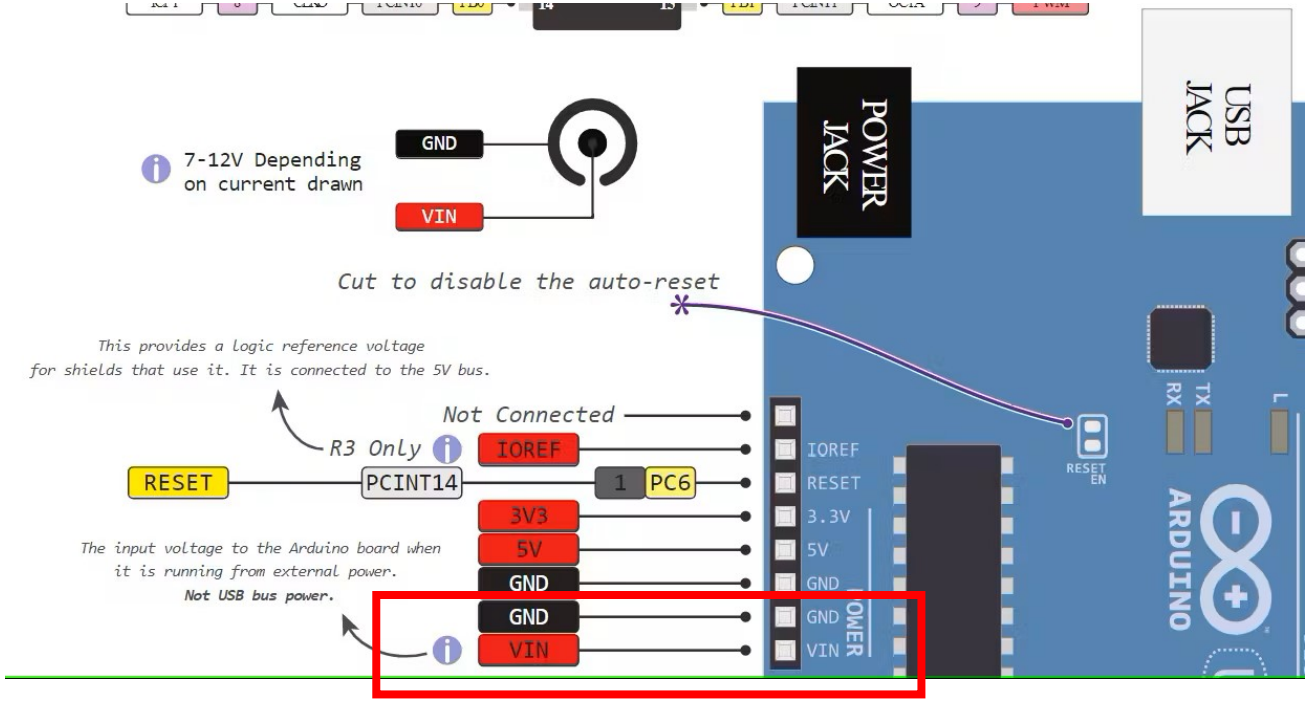
Power Supply

1. Barrel Jack
2. VIN Pin
3. USB cable



[Link]

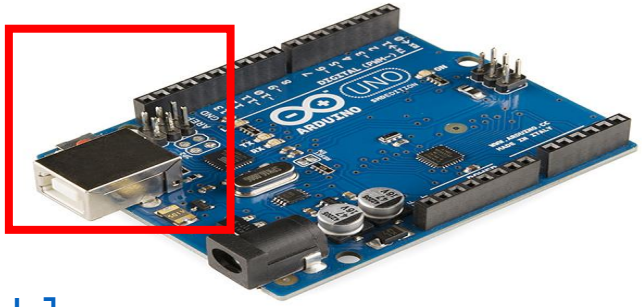
This pin is used to power the Arduino Uno board using an external power source.



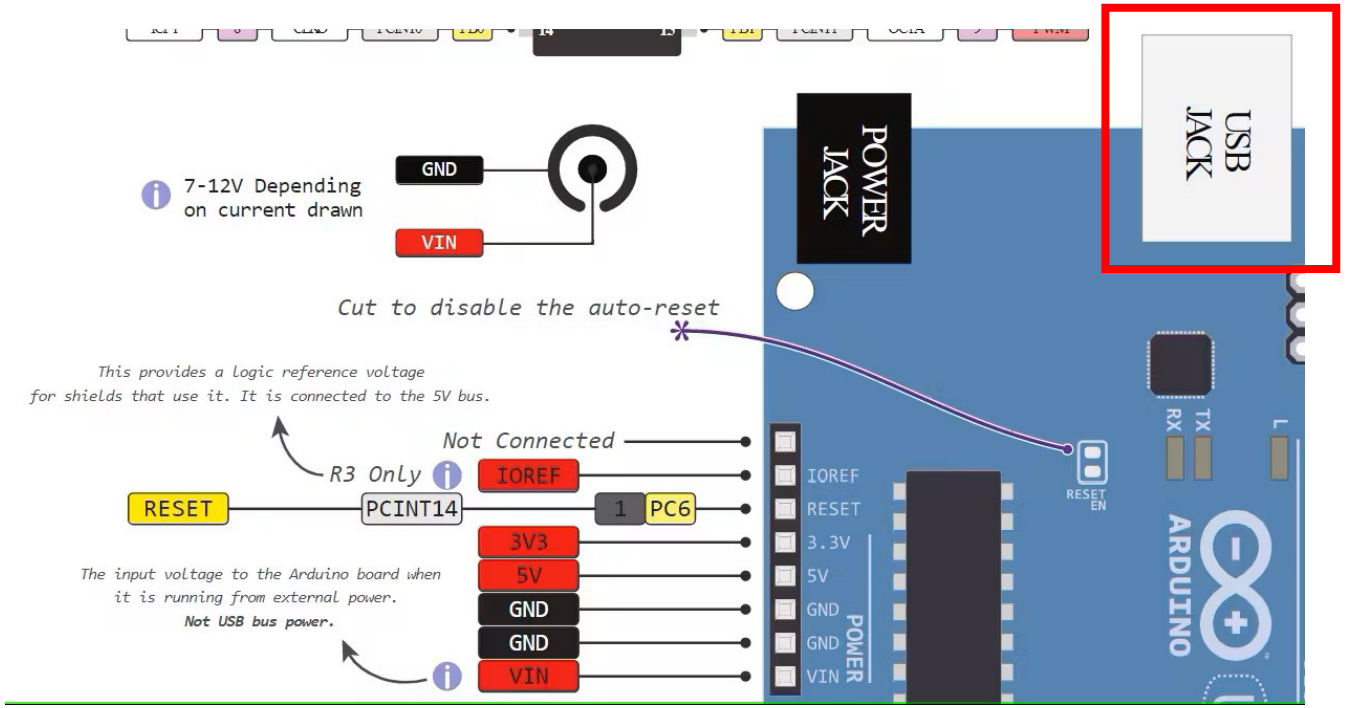
Power Supply

1. Barrel Jack
2. VIN Pin
3. USB cable

When connected to the computer, provides 5 volts at 500mA.



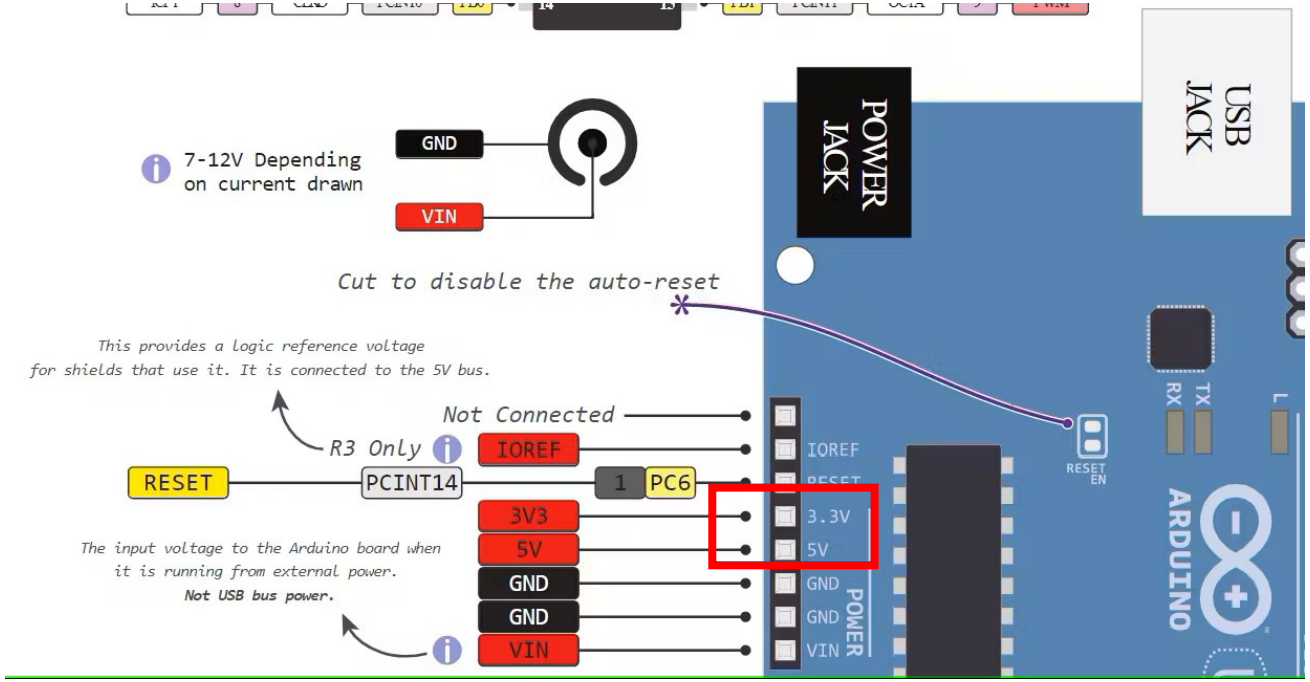
[Link]



Power Supply

4. 5v and 3v3
5. GND
6. RESET
7. IOREF

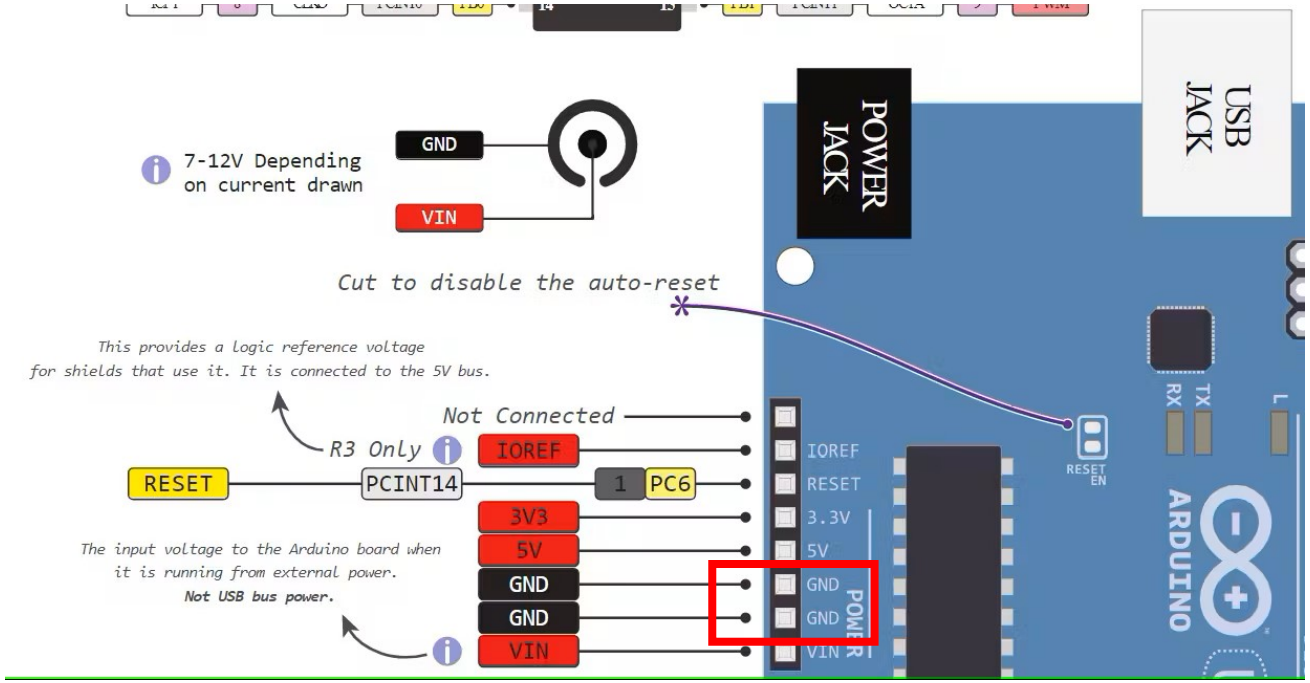
They provide regulated 5v and 3.3v to power external components.



Power Supply

4. 5v and 3v3
5. GND
6. RESET
7. IOREF

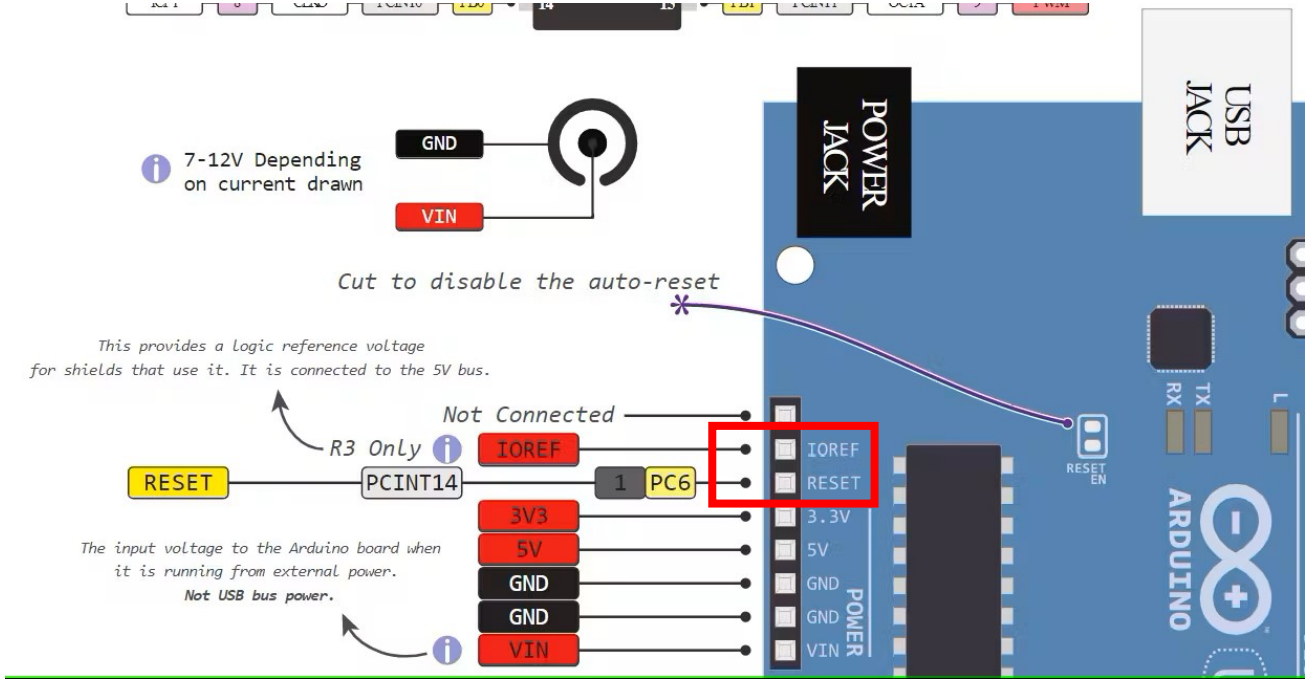
You can find 5 GND pins, which are all interconnected. The GND pins are used to close the electrical circuit and provide a common logic reference level throughout your circuit. Always make sure that all GNDs are connected to one another and have a common ground



Power Supply

4. 5v and 3v3
5. GND
6. RESET
7. IOREF

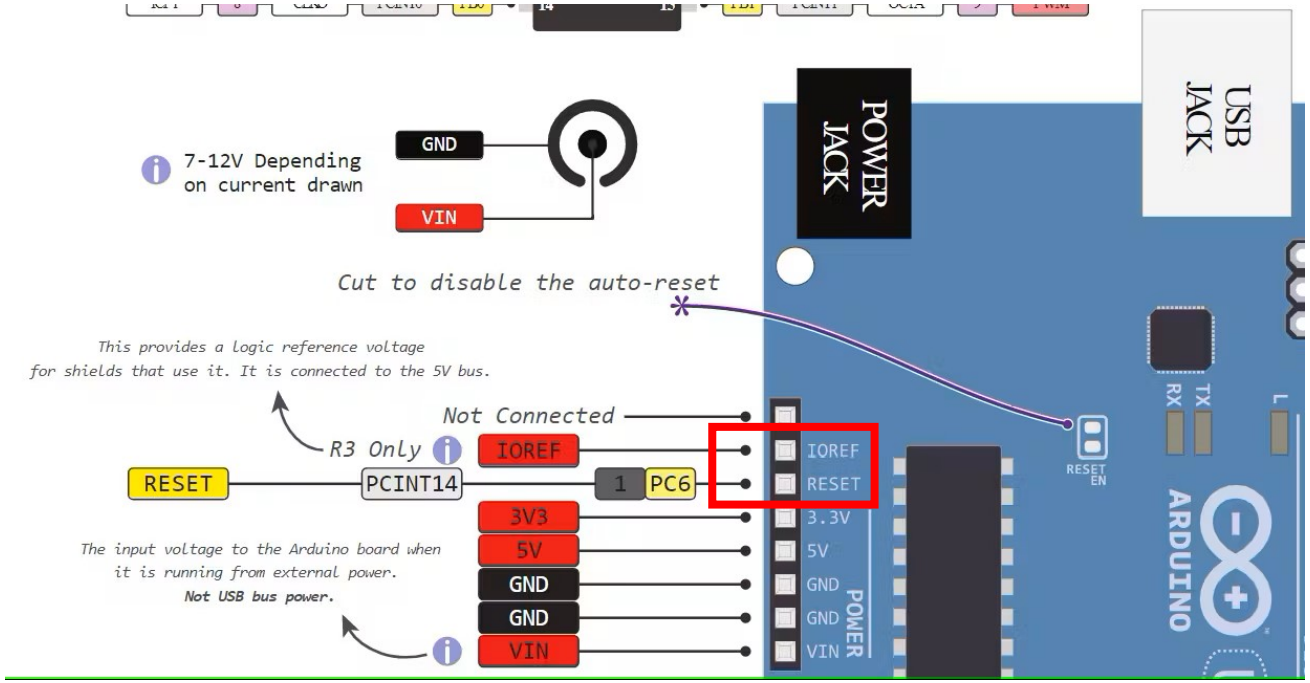
Resets the Arduino.



Power Supply

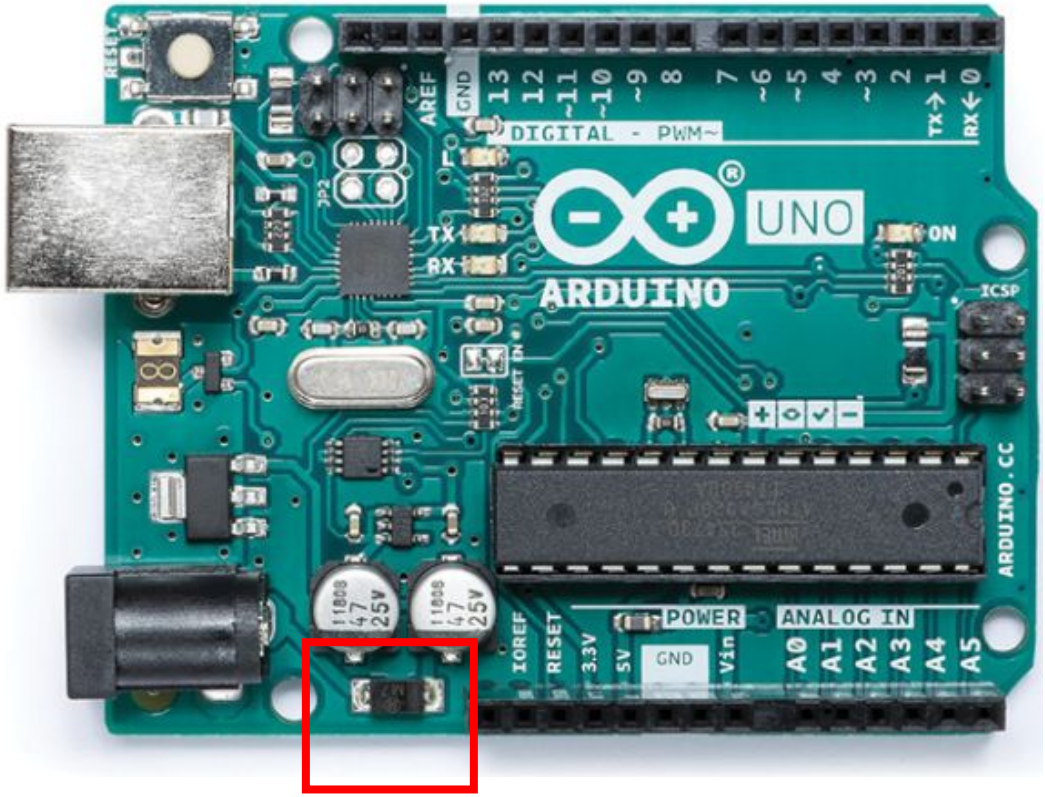
4. 5v and 3v3
5. GND
6. RESET
7. IOREF

This pin is the input/output reference. It provides the voltage reference with which the microcontroller operates.



Power Supply

There is a polarity protection diode connecting between the positive of the barrel jack to the VIN pin, rated at 1 Ampere. [Link]



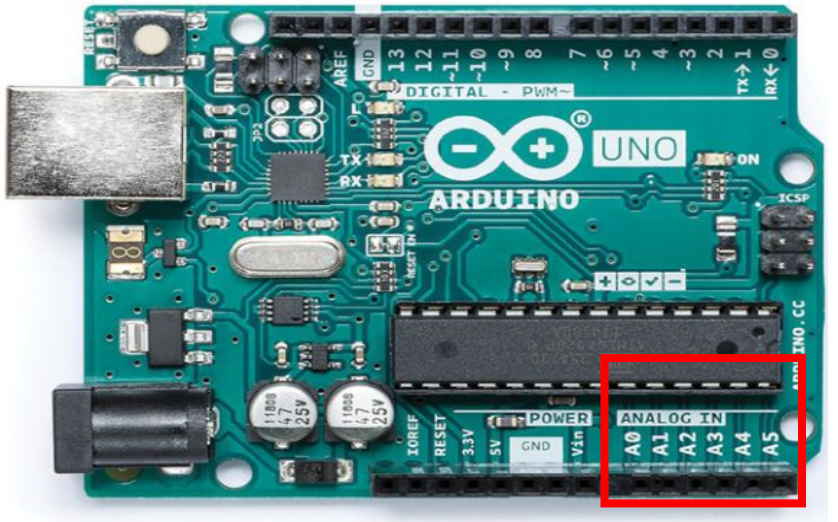
GPIO Pins

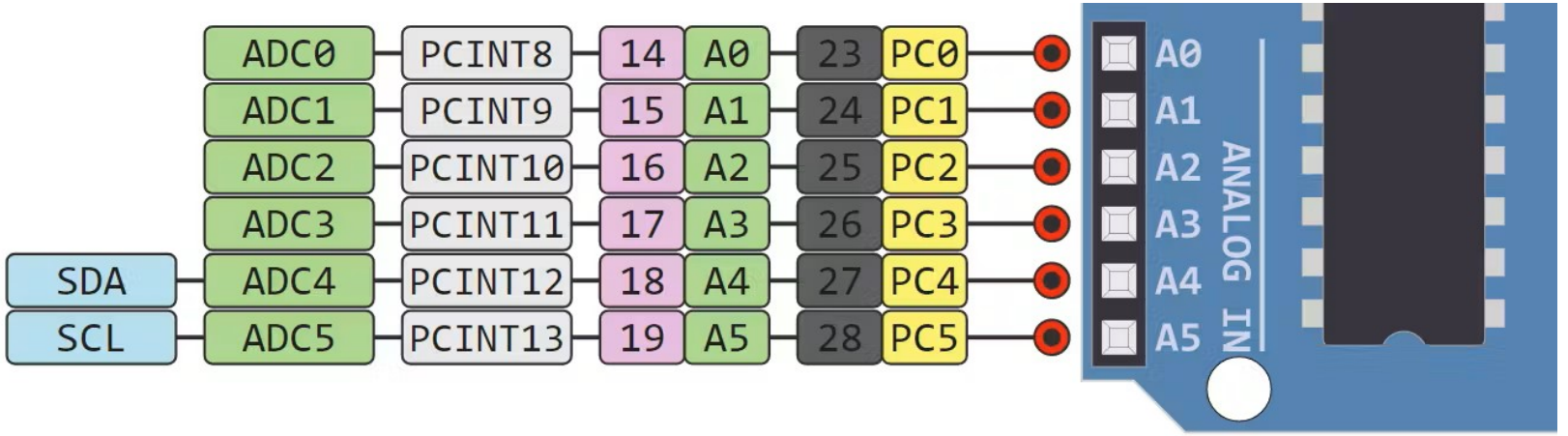
GPIO (General-Purpose Input/Output) pins are digital pins on a microcontroller that can be configured as either input or output. They are used to interface with external devices, sensors, and peripherals, and can be controlled and monitored by software.

Analog Pins

The Arduino Uno has 6 analog pins, which utilize ADC (Analog to Digital converter).

These pins serve as analog inputs but can also function as digital inputs or digital outputs.

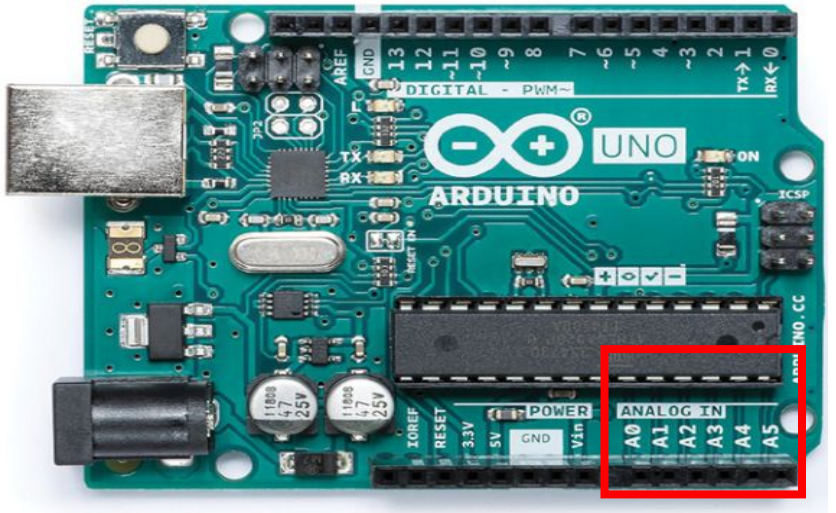


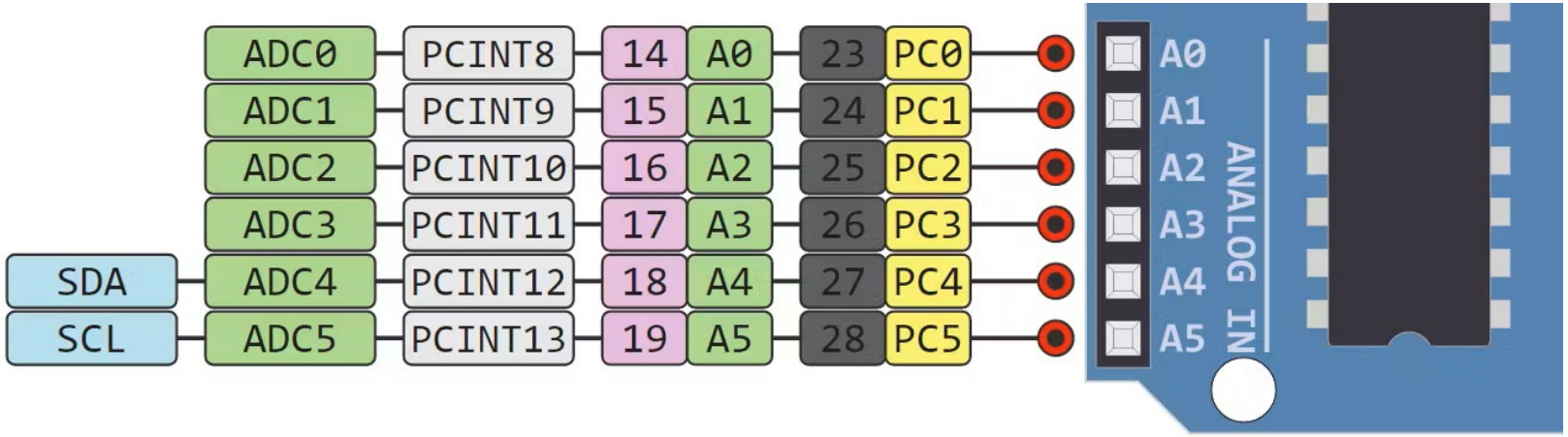


Analog Pins

Analog to Digital Conversion

ADC stands for Analog to Digital Converter. ADC is an electronic circuit used to convert analog signals into digital signals. This digital representation of analog signals allows the processor (which is a digital device) to measure the analog signal and use it through its operation.

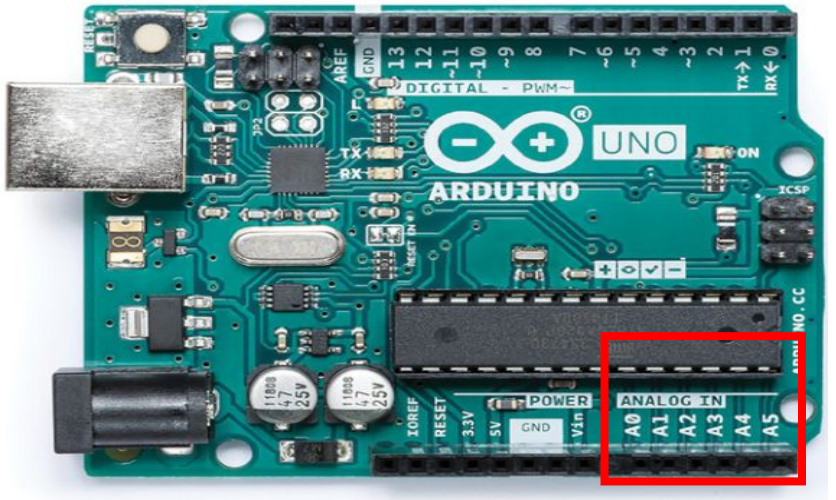


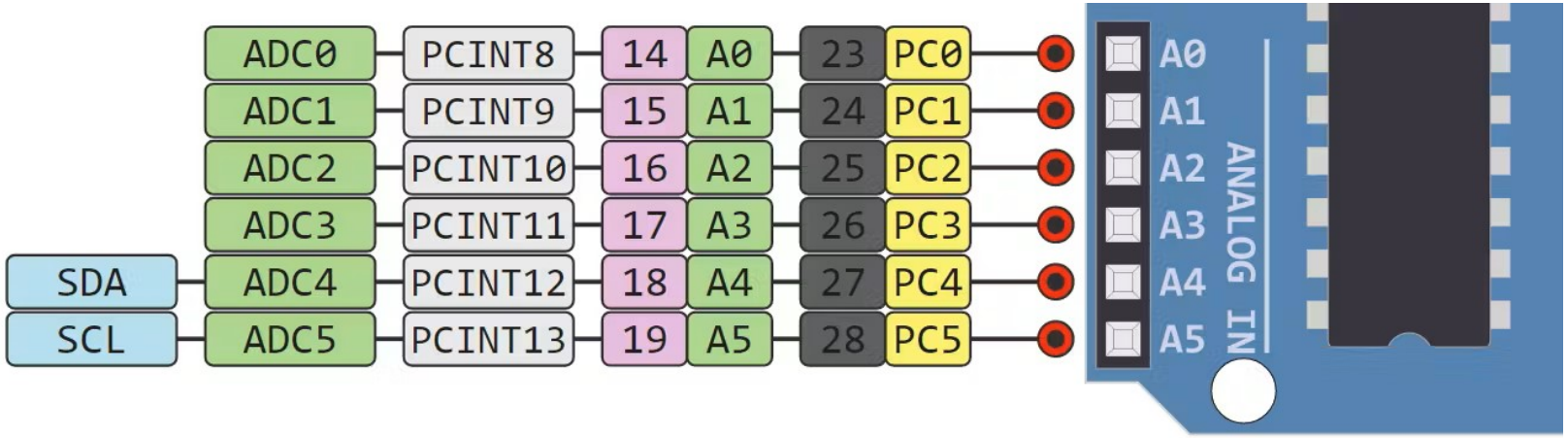


Analog Pins

Analog to Digital Conversion

Arduino Pins A0-A5 are capable of reading analog voltages. On Arduino the ADC has 10-bit resolution, meaning it can represent analog voltage by 1,024 digital levels. The ADC converts voltage into bits which the microprocessor can understand.

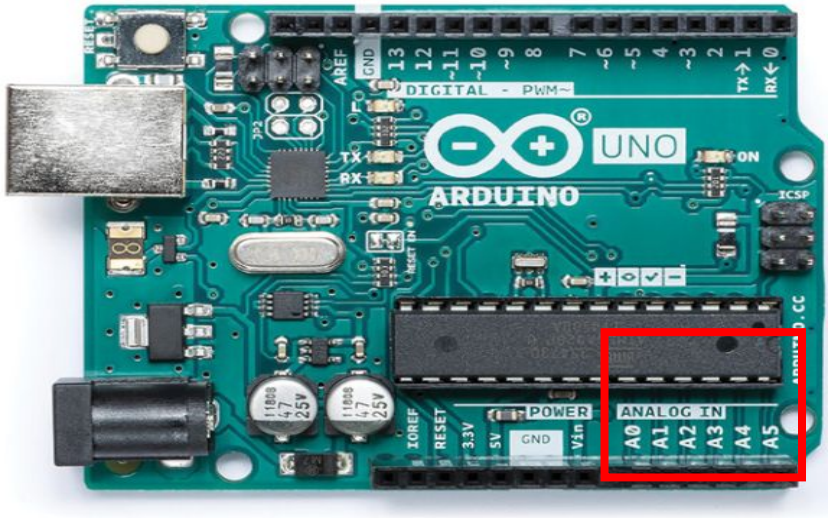


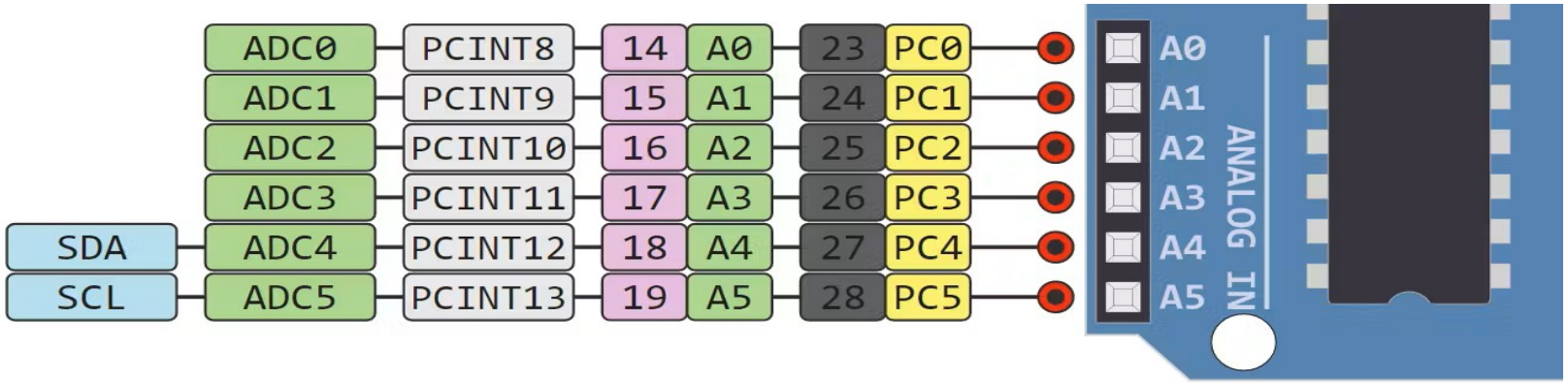


Analog Pins

Analog to Digital Conversion

One common example of an ADC is Voice over IP (VoIP). Every smartphone has a microphone that converts sound waves (voice) into analog voltage. This goes through the device's ADC, gets converted into digital data, which is transmitted to the receiving side over the internet.

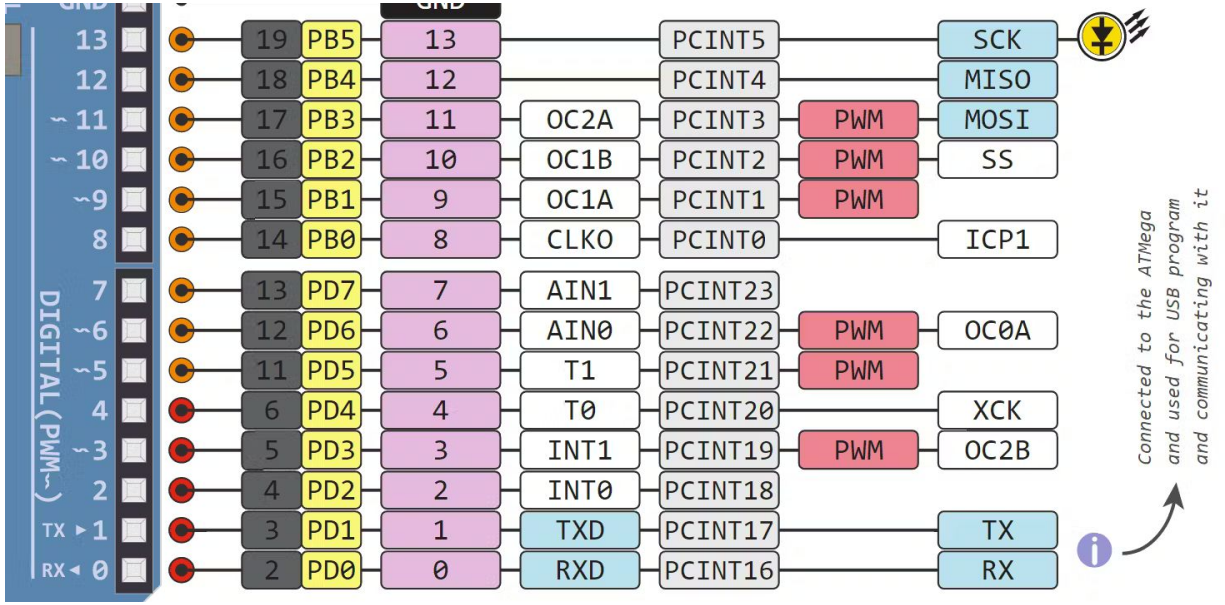




Digital Pins

When the digital pins are configured as output, they are set to 0 or 5 volts.

Pins 0-13 of the Arduino Uno serve as digital input/output pins.

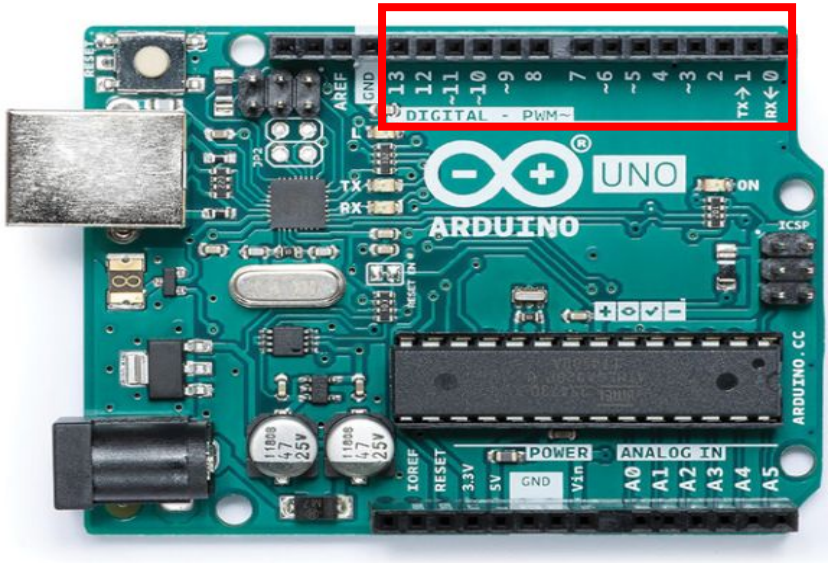


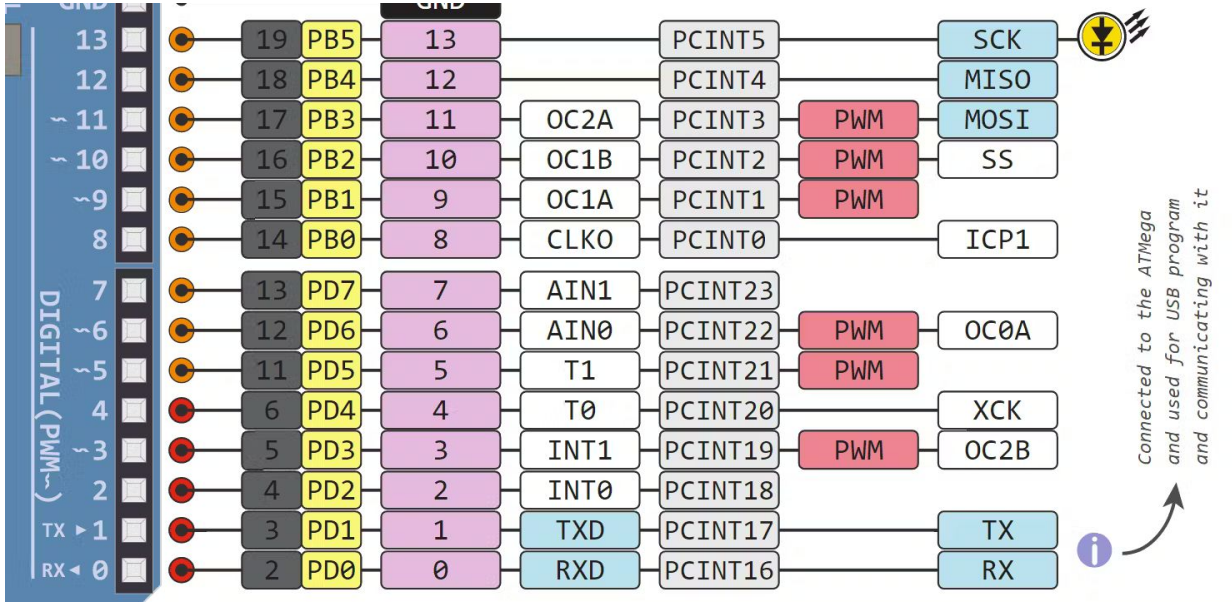
Digital Pins

When the digital pins are configured as input, the voltage is supplied from an external device. This voltage can vary between 0-5 volts which is converted into digital representation (0 or 1). To determine this, there are 2 thresholds:

Below 0.8v - considered as 0.

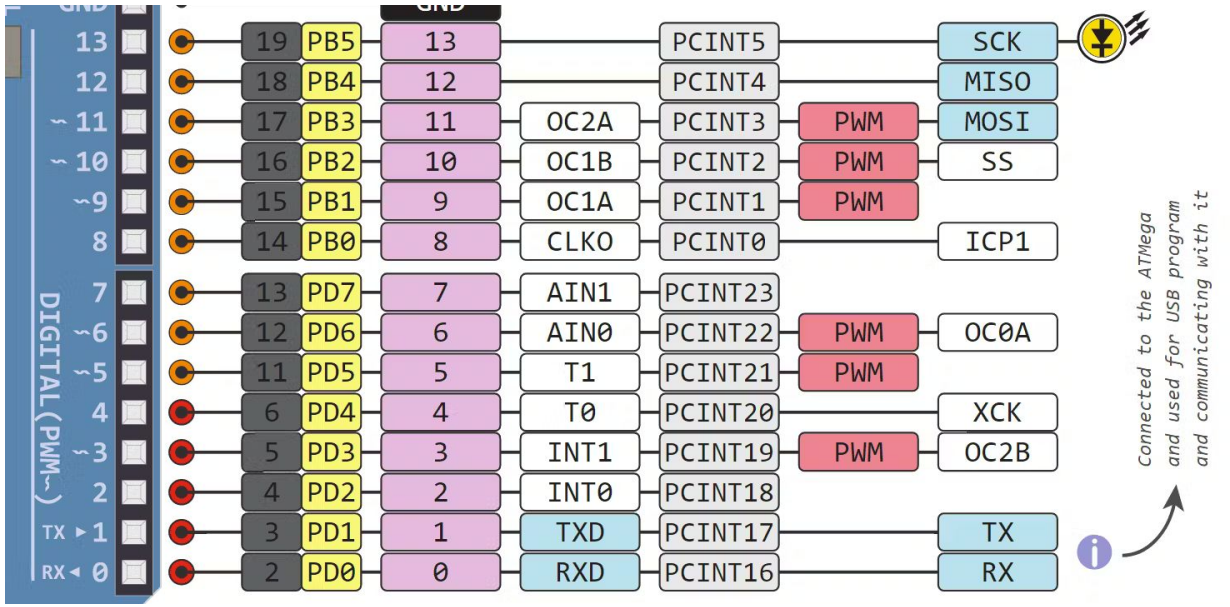
Above 2v - considered as 1.





Digital Pins

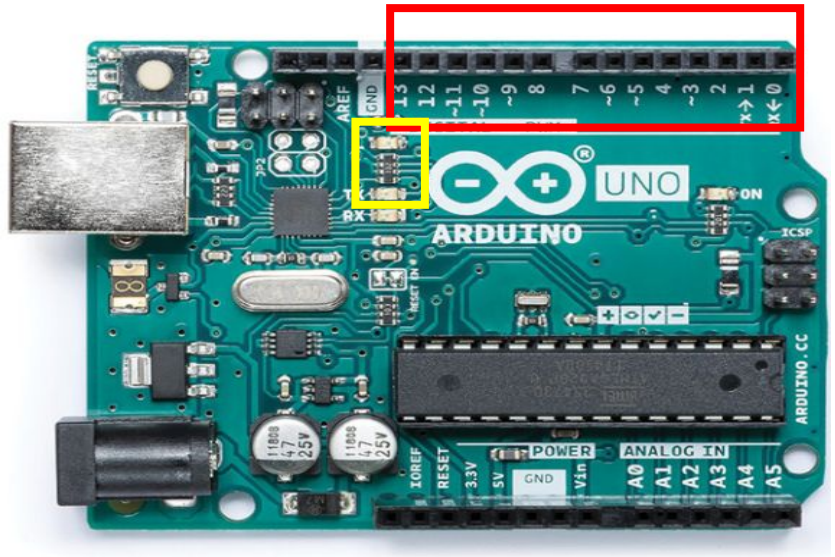
If the voltage is in between the thresholds, the returning value will be undefined.

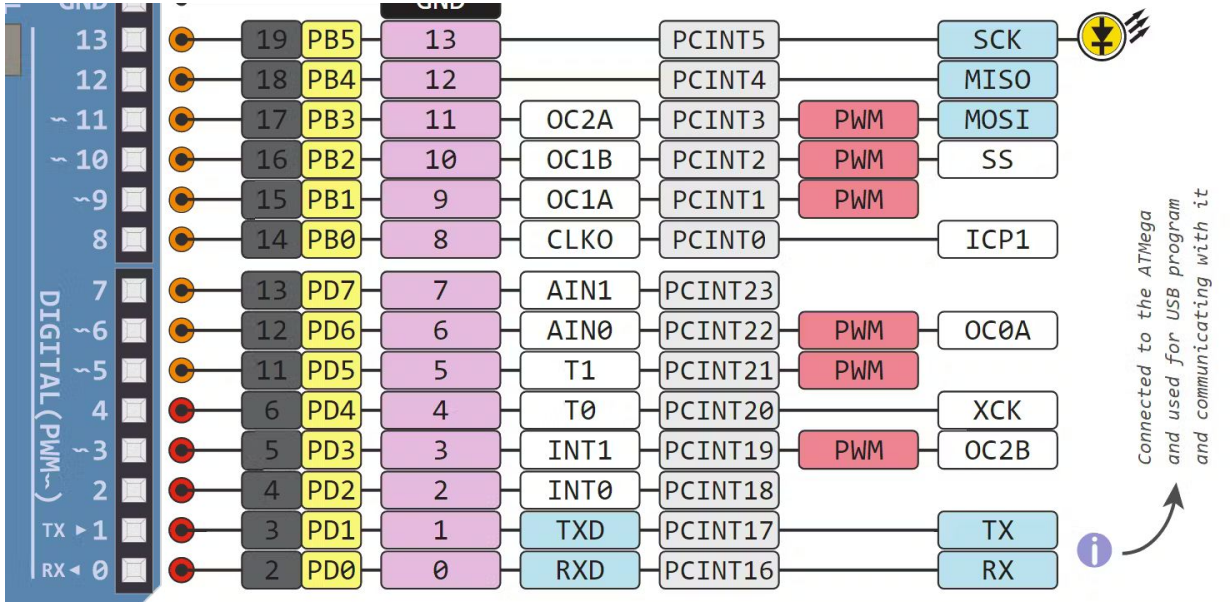


Digital Pins

Pin 13 of the Arduino Uno is connected to the built-in LED.

In the Arduino Uno - pins 3,5,6,9,10,11 have PWM capability.



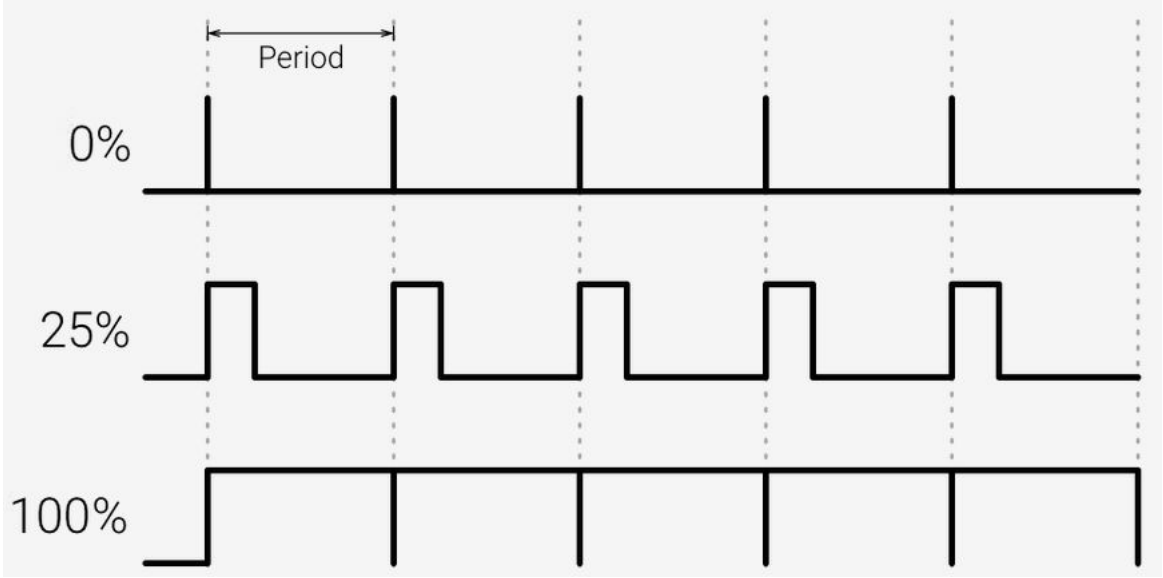


Digital Pins

What is PWM?

In general, Pulse Width Modulation (PWM) is a modulation technique used to encode a message into a pulsing signal.

A PWM is comprised of two key components: frequency and duty cycle.

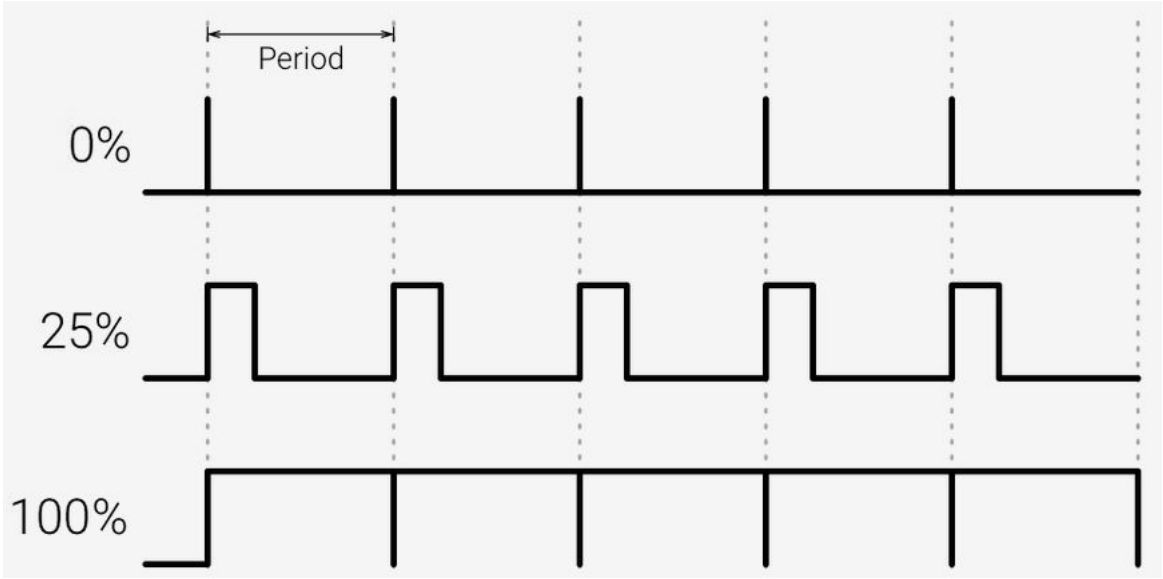


Digital Pins

What is PWM?

The PWM frequency dictates how long it takes to complete a single cycle (period) and how quickly the signal fluctuates from high to low.

The duty cycle determines how long a signal stays high out of the total period. Duty cycle is represented in percentage.

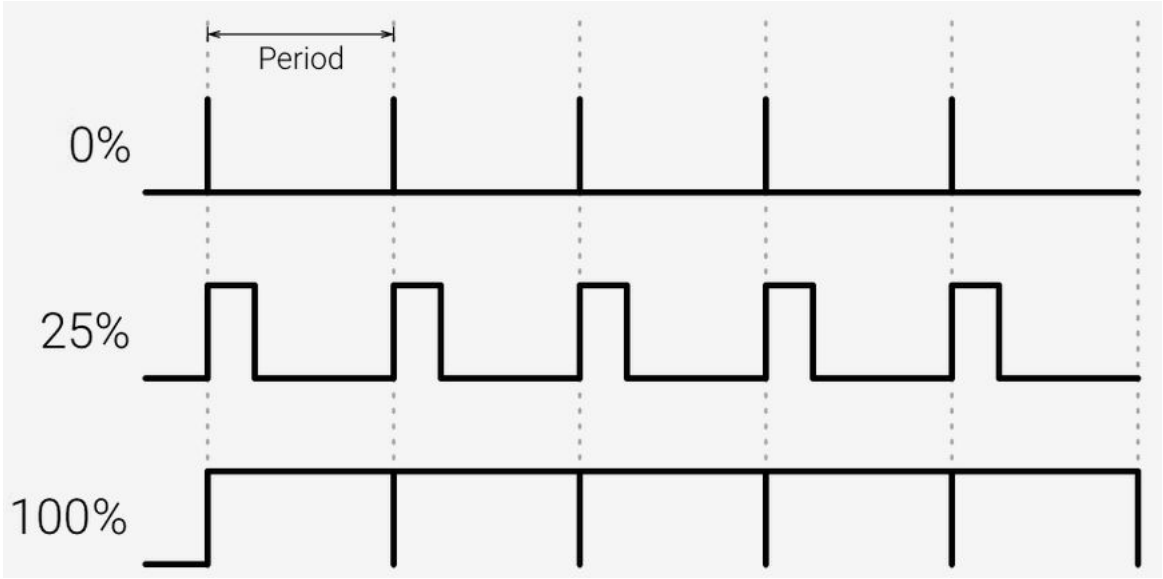


Digital Pins

What is PWM?

In Arduino, the PWM enabled pins produce a constant frequency of ~500Hz, while the duty cycle changes according to the parameters set by the user.

PWM signals are used for speed control of DC motors, dimming LEDs and more.



Digital Pins

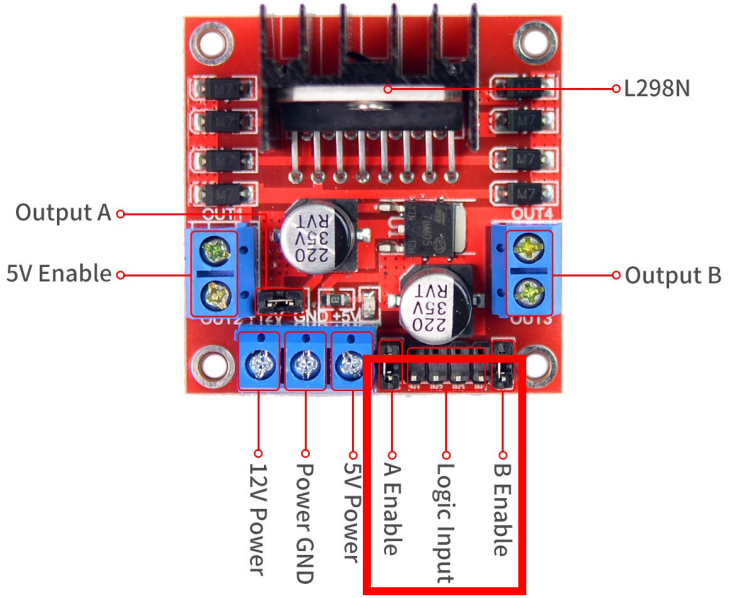
PWM Application – Motor Driver

A constant voltage source powers the motor.

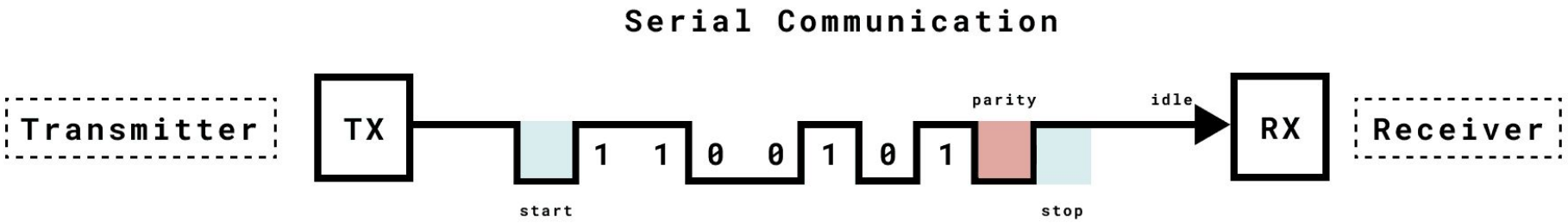
PWM creates a switching square wave signal, alternating between full voltage (on) and zero voltage (off) at a high frequency.

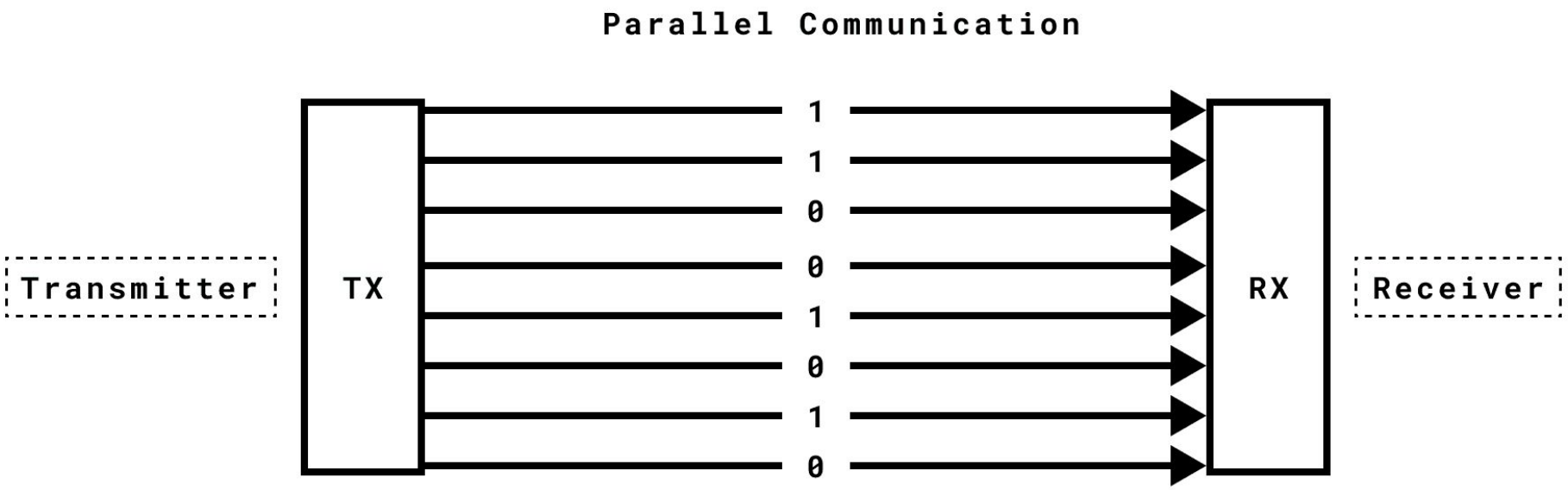
By changing the duty cycle, the on-time compared to the total cycle time, we can control the average voltage experienced by the motor.

[Link]



Serial Communication





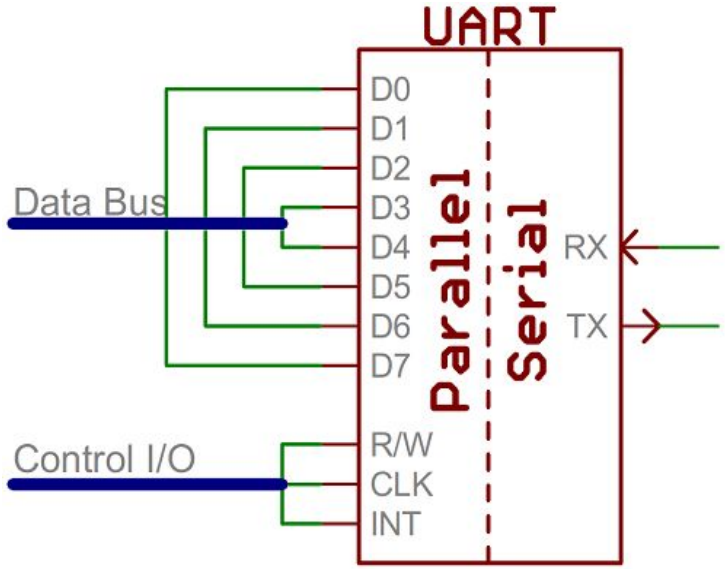
[Link]

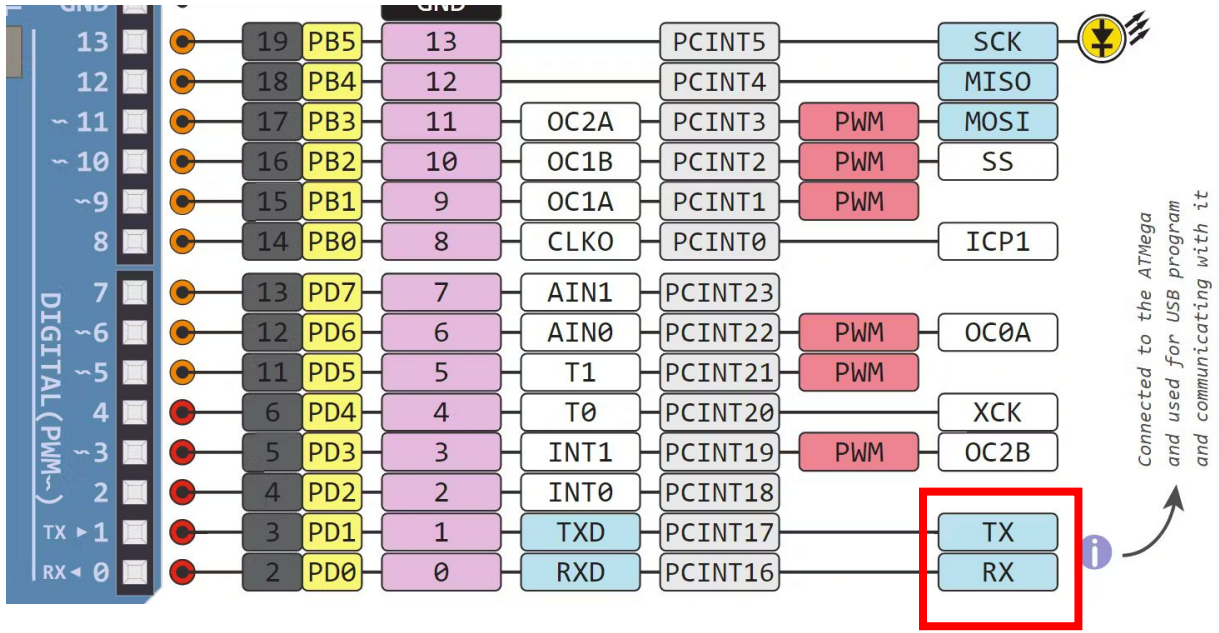
Serial Communication

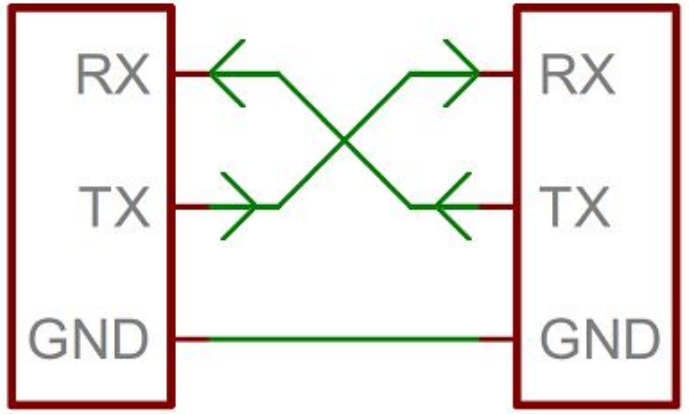
UART (Universal Asynchronous Receiver/Transmitter)

[Link]

A universal asynchronous receiver/transmitter (UART) is a block of circuitry responsible for implementing serial communication. Essentially, the UART acts as an intermediary between parallel and serial interfaces. On one end of the UART is a bus of eight-or-so data lines (plus some control pins), on the other is the two serial wires - RX and TX.







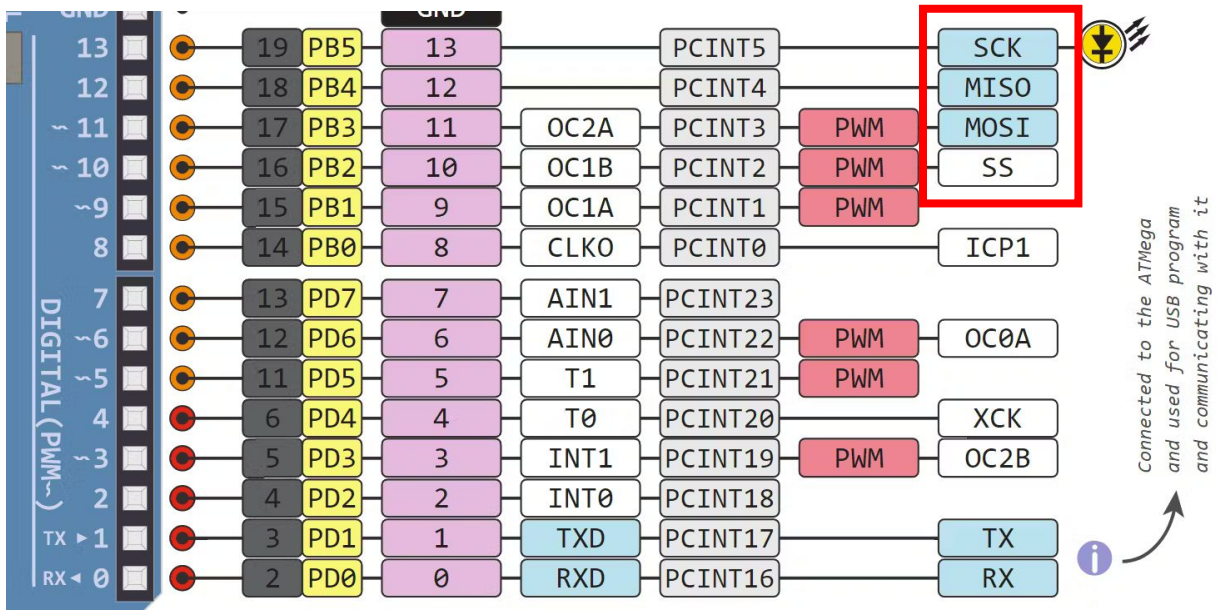
Serial Communication

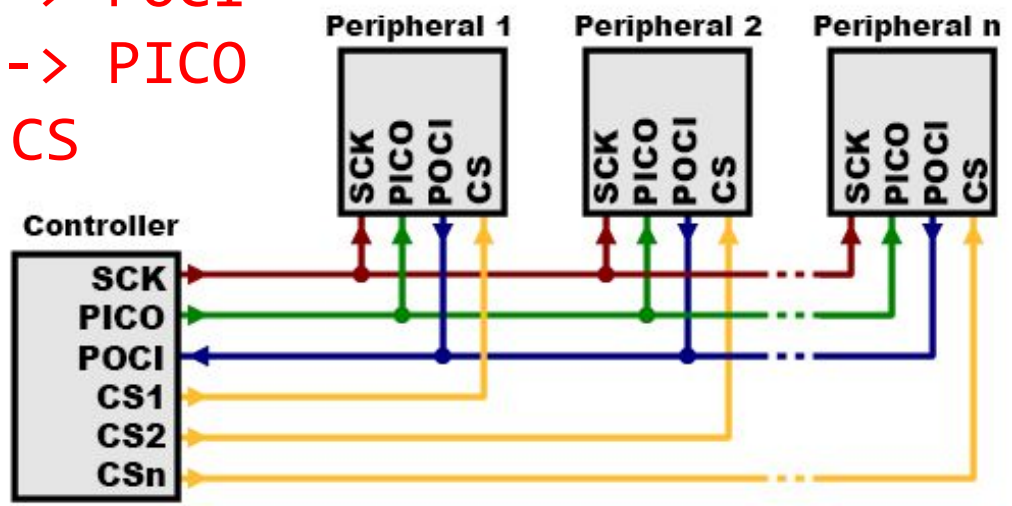
SPI (Serial Peripheral Interface)

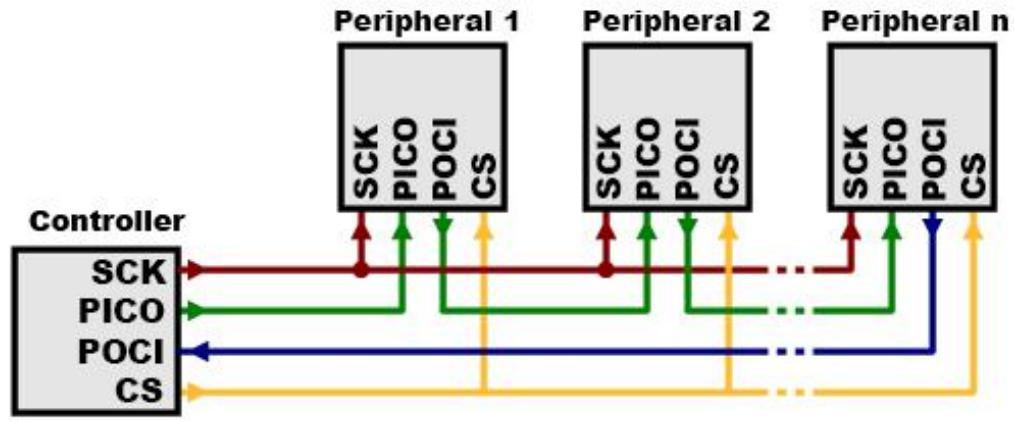
[Link]

Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards. It uses separate clock and data lines, along with a select line to choose the device you wish to talk to.

MISO -> POCI
MOSI -> PICO
SS -> CS







Serial Communication

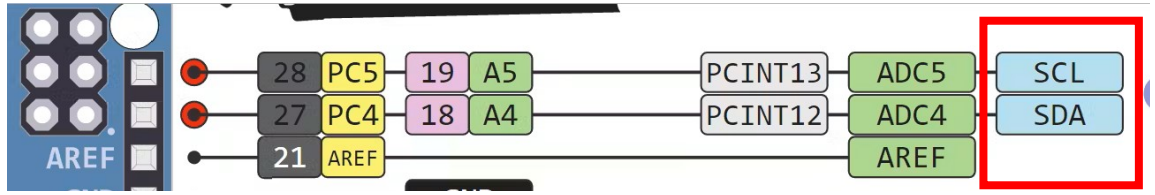
I2C (Inter-Integrated Circuit)

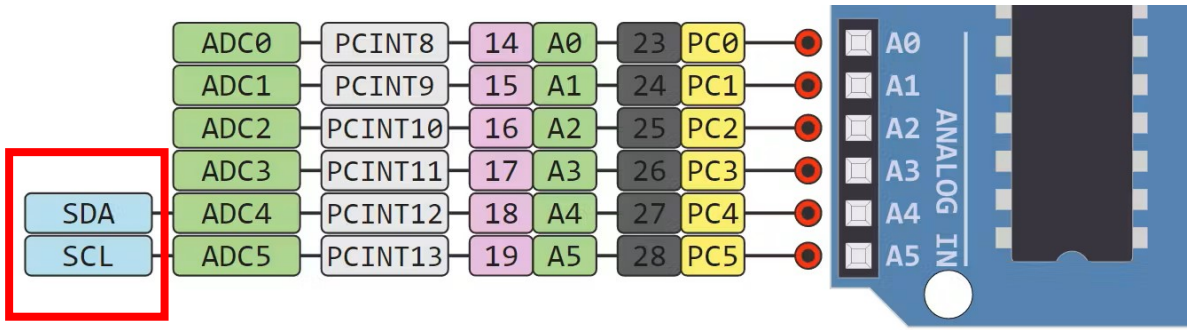
The Inter-Integrated Circuit (I ^{2}C) Protocol is a protocol intended to allow multiple "peripheral" digital integrated circuits ("chips") to communicate with one or more "controller" chips.

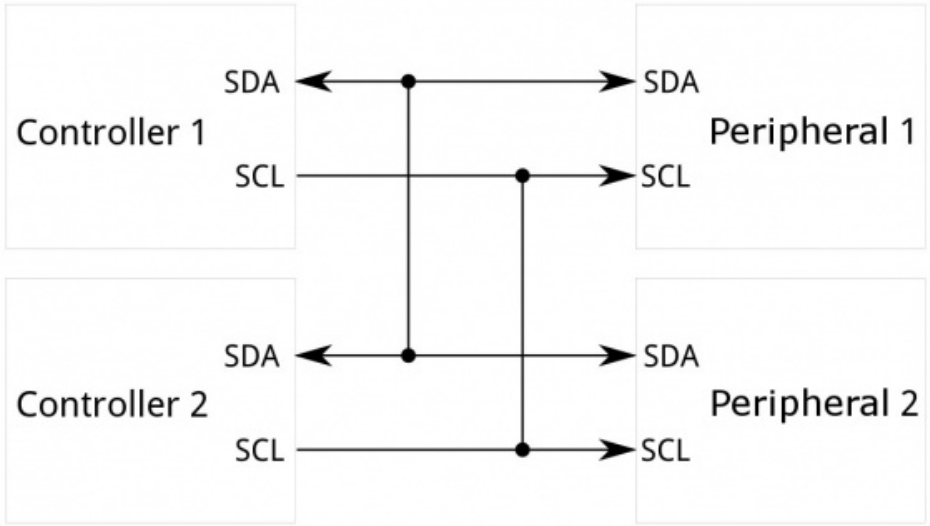
SCL is the clock line which is designed to synchronize data transfers.

SDA is the line used to transmit data.

[Link]







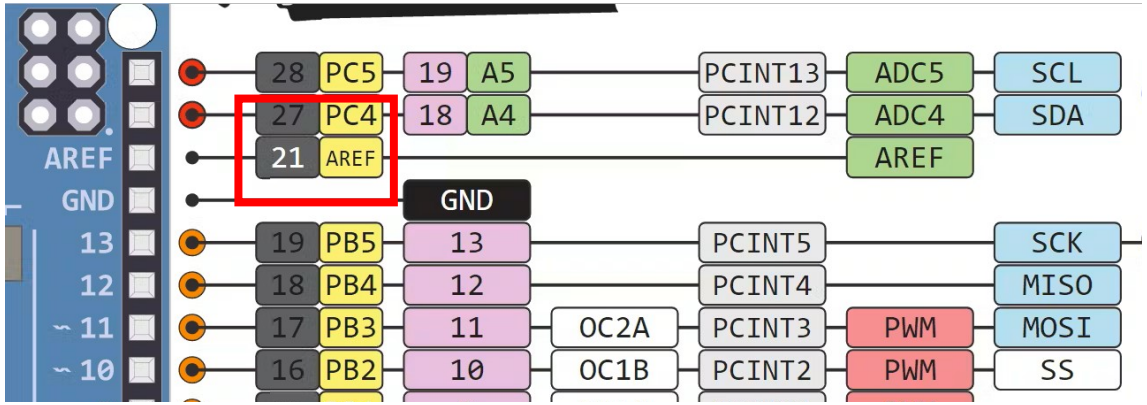
AREF

Reference voltage for the analog inputs. Default 5v.

Arduino ADC is 10 bits (0-1023). So, by default, 5v will be mapped to 1023.

[Link]

$$\frac{ADC\ Resolution}{AREF} = \frac{ADC\ Reading}{Mesured\ Analog\ Voltage}$$



AREF

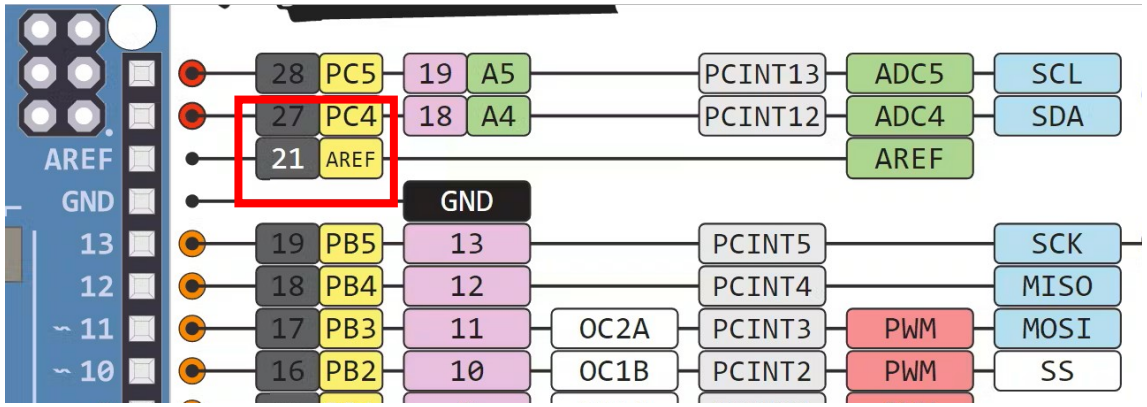
Reference voltage for the analog inputs. Default 5v.

Arduino ADC is 10 bits (0-1023). So, by default, 5v will be mapped to 1023.

[Link]

$$\frac{ADC\ Resolution}{AREF} = \frac{ADC\ Reading}{Mesured\ Analog\ Voltage}$$

$$\rightarrow \frac{1023}{5.0v} = \frac{x}{2.12v}$$



AREF

Reference voltage for the analog inputs. Default 5v.

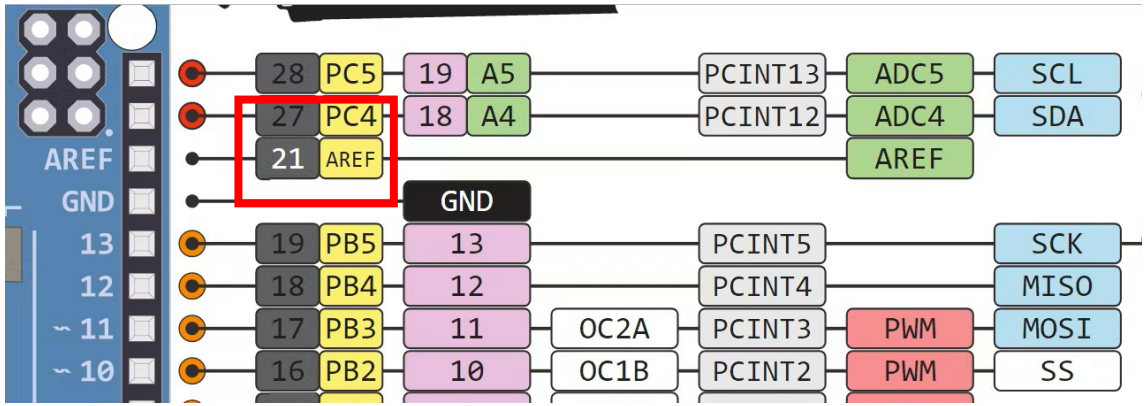
Arduino ADC is 10 bits (0-1023). So, by default, 5v will be mapped to 1023.

[Link]

$$\frac{ADC\ Resolution}{AREF} = \frac{ADC\ Reading}{Mesured\ Analog\ Voltage}$$

$$\rightarrow \frac{1023}{5.0v} = \frac{x}{2.12v}$$

$$\rightarrow x = 434$$

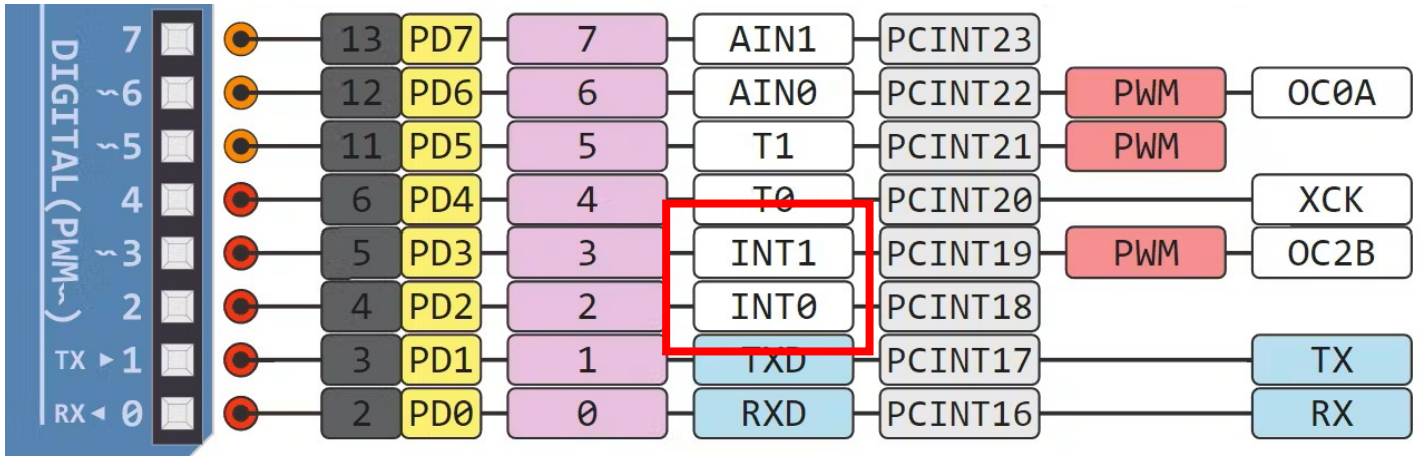


Interrupt

An external interrupt is a system interrupt that occurs when outside interference is present. Interference can come from the user or other hardware devices in the network.

Common uses for these interrupts in Arduino are reading the frequency a square wave generated by encoders or waking up the processor upon an external event.

[Link]

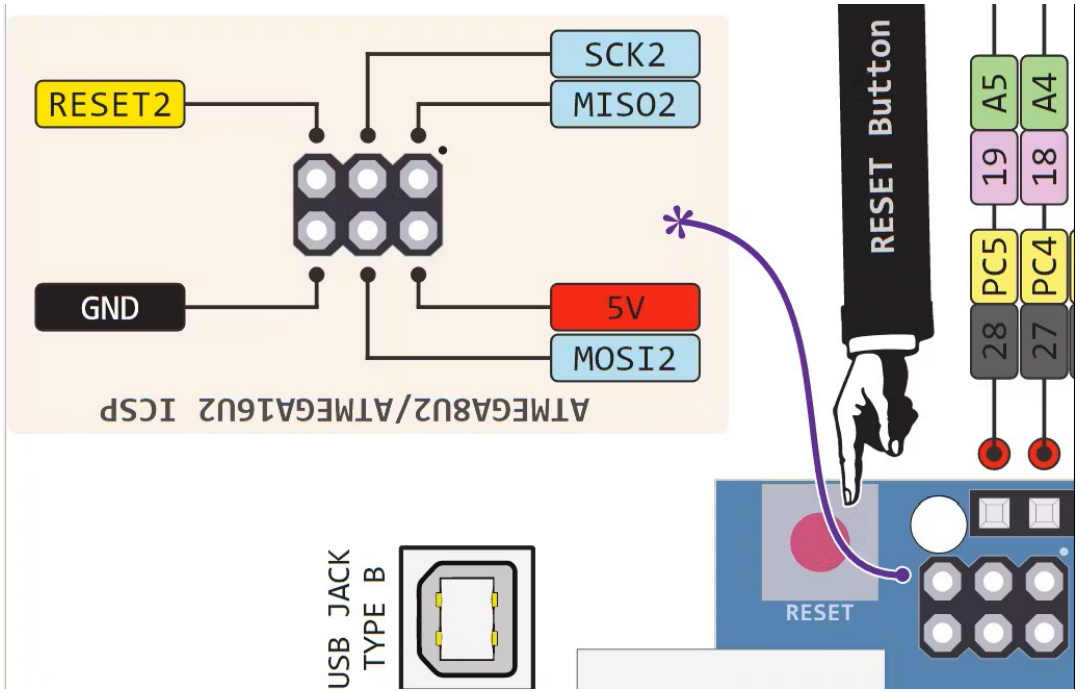


ICSP Header

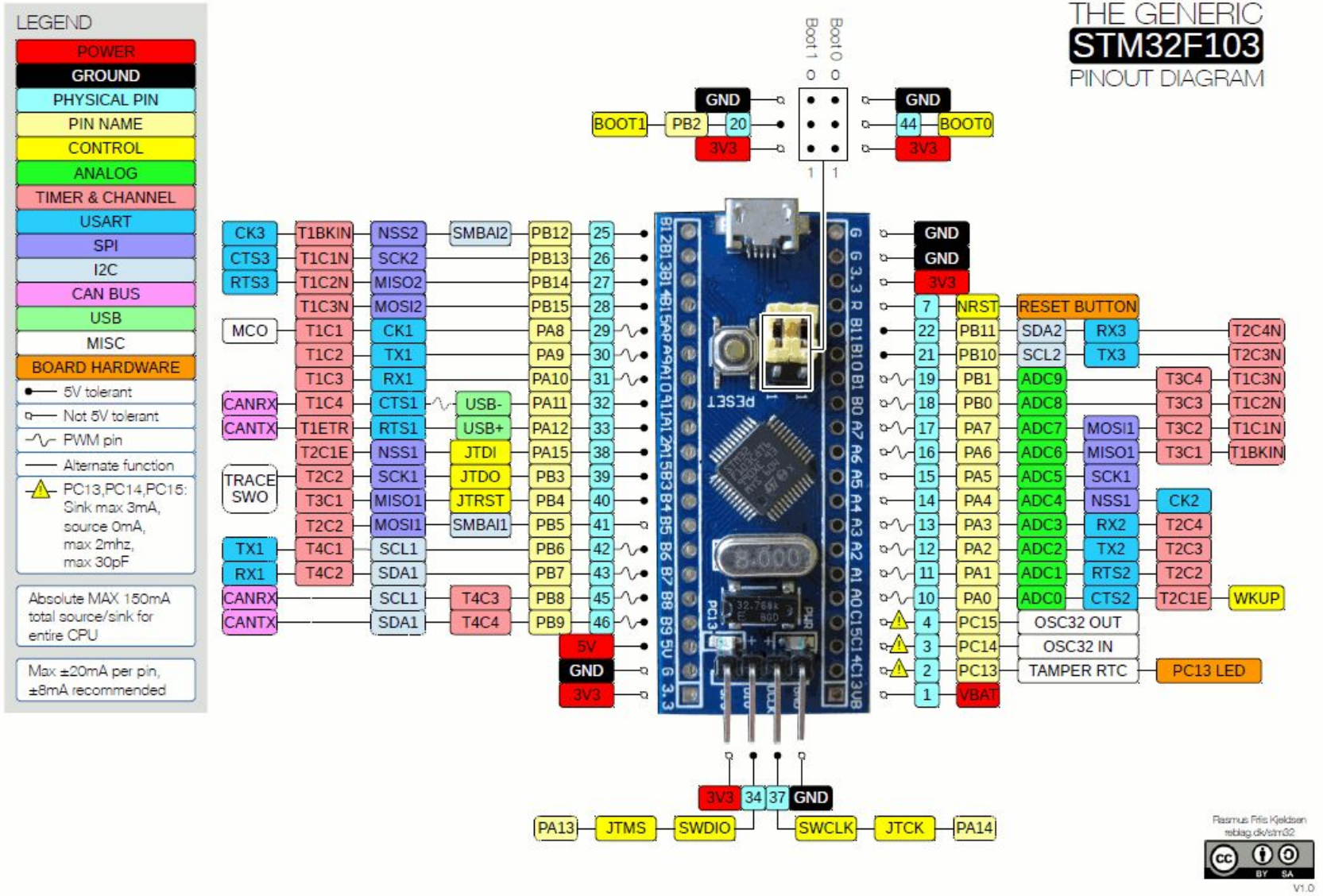
In-Circuit Serial Programming

These pins enable the user to program the Arduino boards' firmware.

[Link]



Arduino Uno vs STM32 Blue Pill



Some production grade usage—

- Industrial Automation Systems
- 3D Printer Controller
- Unmanned Aerial Vehicles (UAVs)
- High-Fidelity Motor Control
- Internet of Things (IoT) Devices

[Link]

Arduino Uno vs STM32 Blue Pill

Functionality	Arduino	STM32 Blue Pill
Architecture	8-bit AVR ATmega328P designed by ATMEL (acquired by Microchip)	32-bit ARM Cortex M3 designed by ARM (Part No: STM32F103C8T6)
Manufacturer	Microchip/ATMEL	ST-Microelectronics
CPU Frequency	16 MHz	72 MHz
Flash Memory	32KB	64KB
RAM	2KB	20KB

Arduino Uno vs STM32 Blue Pill

Functionality	Arduino	STM32 Blue Pill
No of GPIO pins	14 (Digital)	37
No of Analog input Pins	6 (10-bit)	10 (12-bit)
No of PWM pins	6	12
No of Timers	2 (8-bit), 0 (PWM)	3 (16-bit), 1 (PWM)

Thank You!