Name: Shubham Lad

Class: MSc Computer Science (Part 1)

Roll No: 512

Semester: II

Subject: Advanced Embedded Systems

Topic: LED Blinking System (Mini Project)

Academic Year: 2022-2023

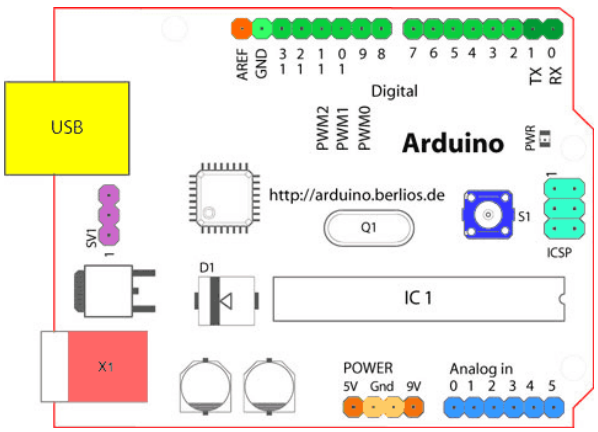
Introduction

Arduino

Arduino comprises of both a physical programmable circuit board (commonly known as a microcontroller) and a programming software, or IDE (Integrated Development Environment) that can be run on a PC, used to compose and transfer PC code to the circuit board. It can be done by using the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Unlike other programmable circuit boards, the Arduino does not require a different equipment (called a software engineer) to upload code to the circuit board, one can essentially utilize a USB link. Also, the Arduino IDE utilizes a rearranged rendition of C++, making it simpler to figure out how to program. In a word, Arduino make the functions of the micro-controller into a more accessible package. The Uno is one of the more prevalent boards in the Arduino family and an extraordinary option for the beginners.

Common Components of Arduino Boards

There are different types of Arduino boards for different purposes. But all the boards have the majority of following components in common.



Starting clockwise from the top center:

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (digitalRead and digitalWrite) if serial communication is also being used (e.g. Serial.begin)
- Reset Button - S1 (dark blue)

- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

Digital Pins

The digital pins on an Arduino board can be used for general purpose input and output via the pinMode(), digitalRead(), and digitalWrite() commands. Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input.

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
- 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.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.

Analog Pins

The analog input pins support 10-bit analog-to-digital conversion (ADC) using the analogRead() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19.

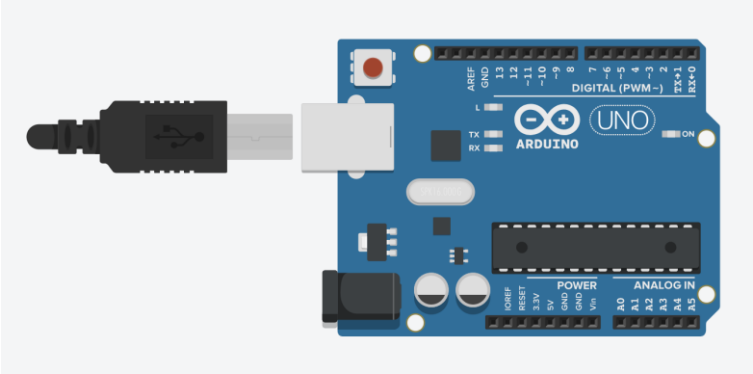
Power Pins

- 9V: 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). Different boards accept different input voltages ranges.
- 5V: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3: (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip.
- GND: Ground pins.

Description of Major Components used

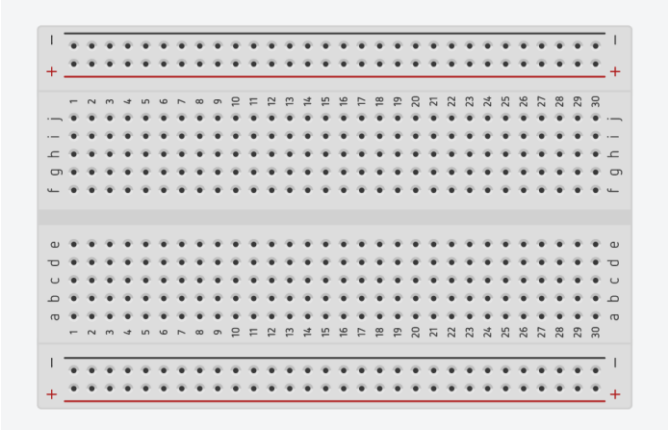
Arduino UNO

The Uno is one of the most popular Arduino boards. It consists of 14-digital I/O pins, where 6-pins can be used as PWM(pulse width modulation outputs), 6-analog inputs, a reset button, a power jack, a USB connection and more. It includes everything required to hold up the microcontroller; simply attach it to a PC with the help of a USB cable and give the supply to get started with a AC-to-DC adapter or battery.



Breadboard

A breadboard is a construction base for prototyping of electronics. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term “breadboard” is commonly used to refer to these. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

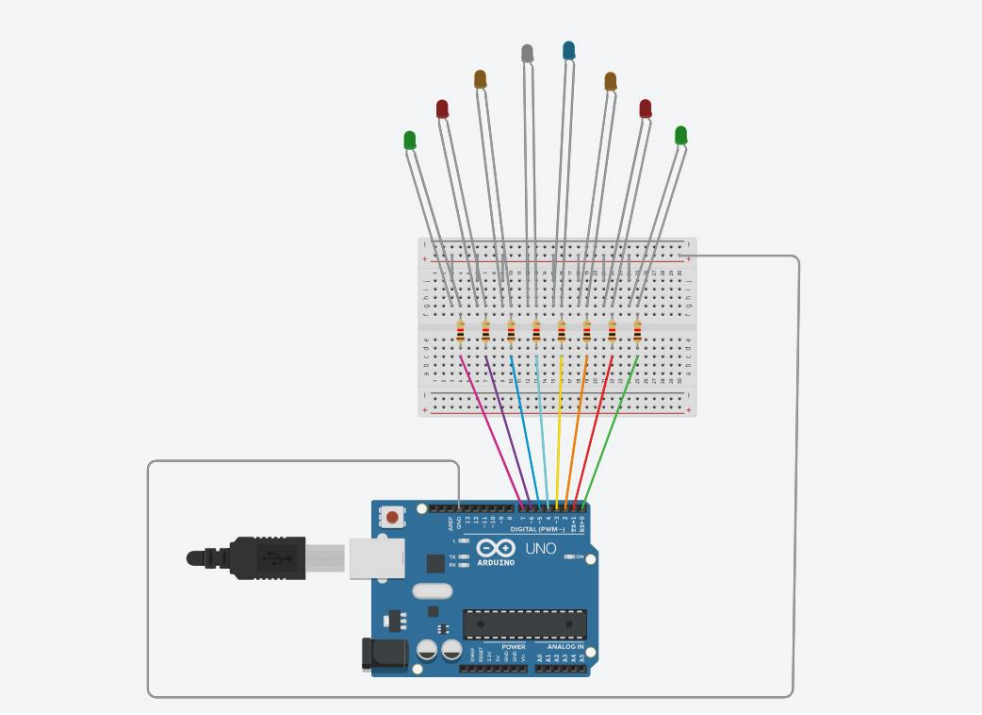


Arduino Software

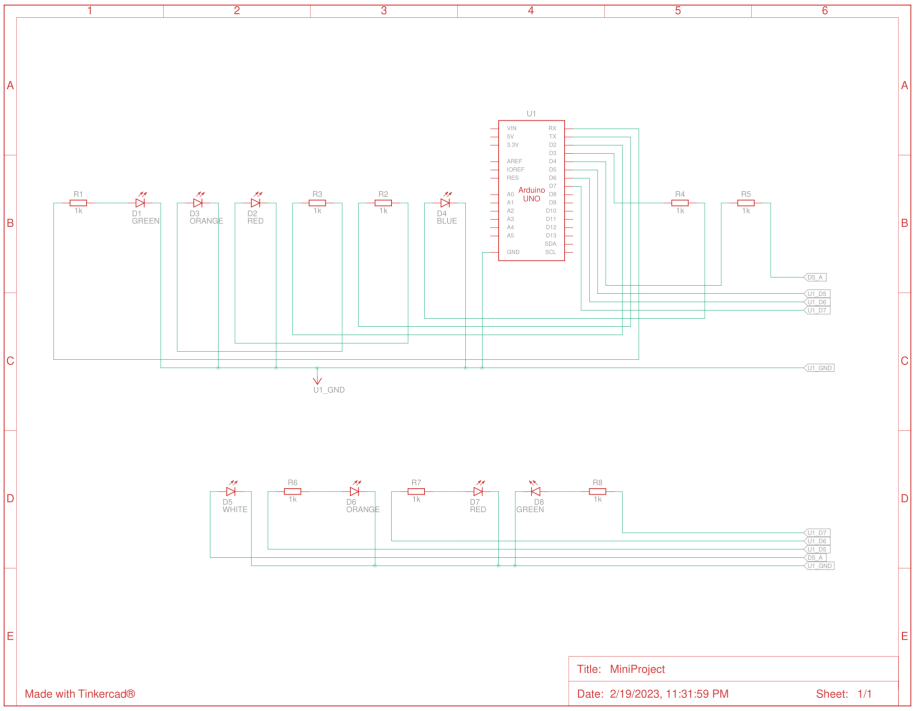
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.

Circuit Setup

- First, ground is connected to the breadboard through the Arduino.
- Followed by connecting the LEDs to the breadboard through male-female wires and Arduino pins to the breadboard.
- Resistors are placed in between these connections to control how much electricity flows through the LED and therefore how brightly it shines.



Graphical Representation



Circuit Schematic

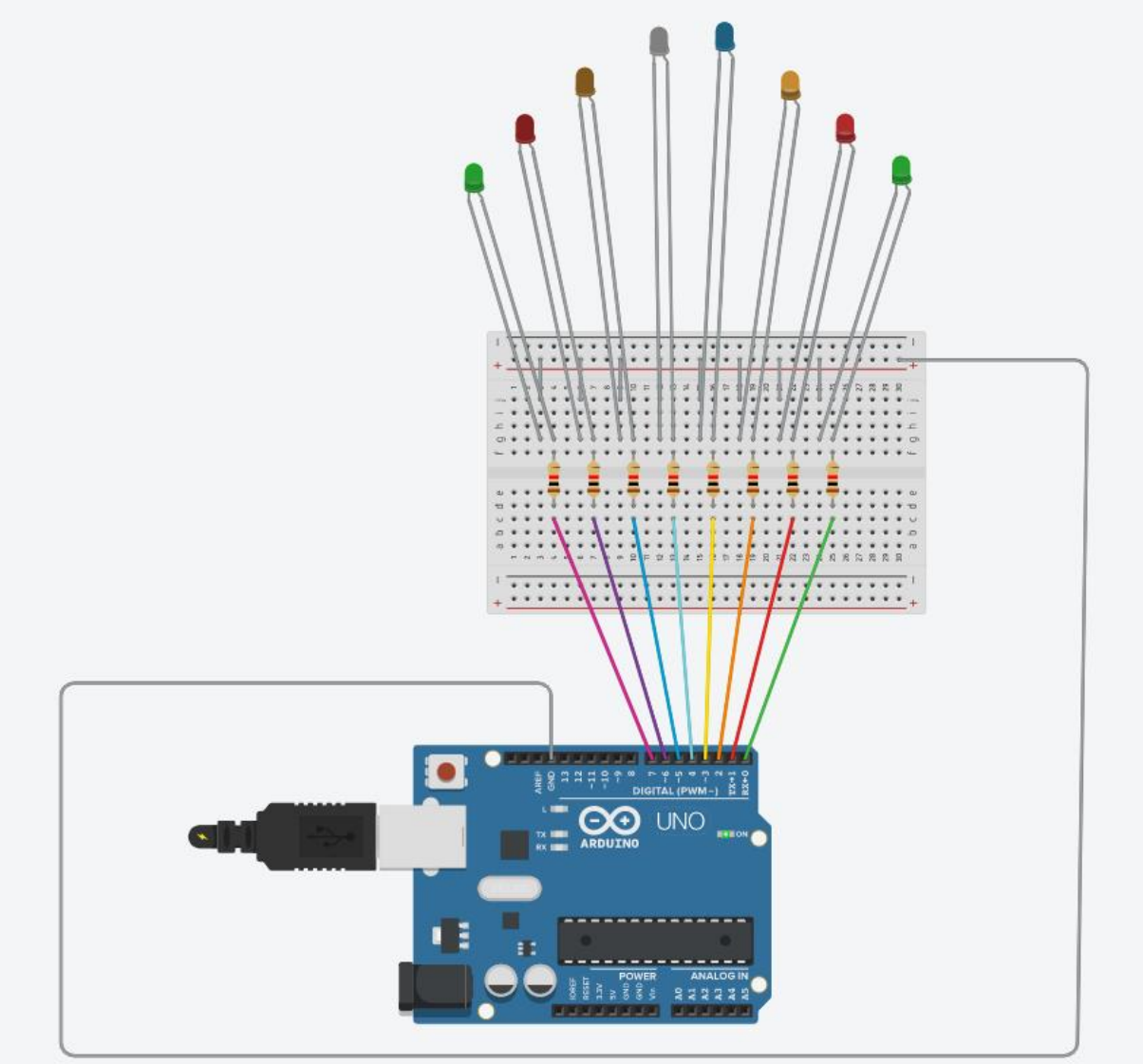
Arduino Code

```
int led1 = 7;
int led2 = 6;
int led3 = 5;
int led4 = 4;
int led5 = 3;
int led6 = 2;
int led7 = 1;
int led8 = 0;
void setup()
{
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
    pinMode(led4, OUTPUT);
    pinMode(led5, OUTPUT);
    pinMode(led6, OUTPUT);
    pinMode(led7, OUTPUT);
    pinMode(led8, OUTPUT);
```

```
}
void loop()
{
    digitalWrite(led8, HIGH);
    delay(50);
    digitalWrite(led8, LOW);
    delay(50);
    digitalWrite(led7, HIGH);
    delay(50);
    digitalWrite(led7, LOW);
    delay(50);
    digitalWrite(led6, HIGH);
    delay(50);
    digitalWrite(led6, LOW);
    delay(50);
    digitalWrite(led5, HIGH);
    delay(50);
    digitalWrite(led5, LOW);
    delay(50);
    digitalWrite(led4, HIGH);
    delay(50);
    digitalWrite(led4, LOW);
    delay(50);
    digitalWrite(led3, HIGH);
    delay(50);
    digitalWrite(led3, LOW);
    delay(50);
    digitalWrite(led2, HIGH);
    delay(50);
    digitalWrite(led2, LOW);
    delay(50);
    digitalWrite(led1, HIGH);
    delay(50);
    digitalWrite(led1, LOW);
    delay(50);
}
```

Observation

On starting the simulation the LEDs starts blinking one after another with a time interval of 50ms (0.05 seconds).


DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN