Arduino Reference Guide: Libraries, LEDs, and Other References

Anna Yrjanson

Arduino Programming Basics

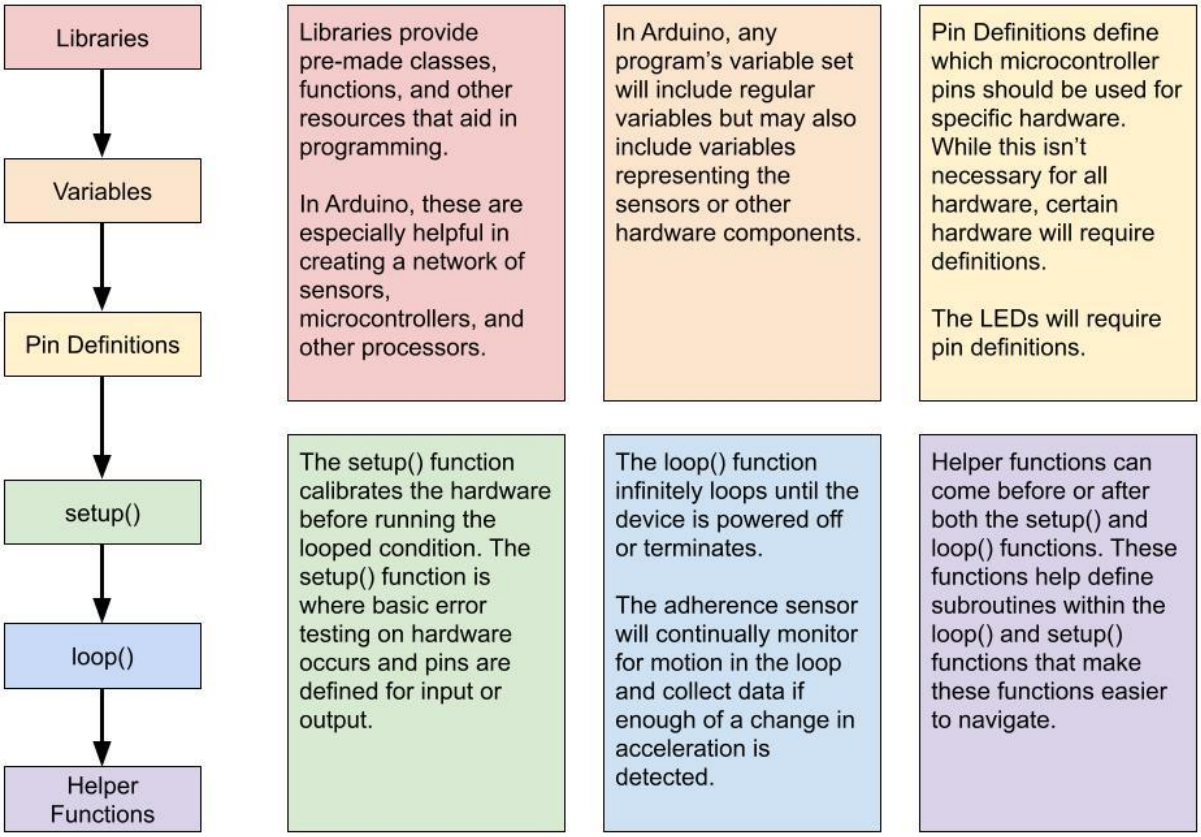

Figure 1 Arduino Program Flow Diagram

Figure 1 above displays the order of code in a standard Arduino program. This order is applicable for the adherence sensor program as well, the previous team used a similar structure for their device.

Library Source: https://www.heavy.ai/technical-glossary/open-source-library

Sensors, Microcontrollers, and Libraries

SparkFun Triple Axis Accelerometer Breakout (KX134)

This accelerometer requires the "SparkFun XK13X Arduino Library" which configures the accelerometer and allows Arduino, ESP32 and accelerometer to communicate with each other.

Documentation: https://www.arduino.cc/reference/en/libraries/sparkfun-kx13x-arduino-library/

Notable Structs and Functions:

SparkFun_KX134 – creates an accelerometer object

- SparkFun_KX134 kxAccel; // Creates a new accelerometer object
- kxAccel.begin(); // Starts communication with the accelerometer
- kxAccel.enableDataEngine(); // Enables a bit that indicates that data is ready
- ksAccel.dataReady(); // Used in conditional statements to check if the data is ready

outputData – creates an outputData object, this accesses the accelerometer's data

- xkAccel.getAccelData(&myData) // De-references the data on the accelerometer
- Serial.print(myData.xData); // Prints the x-axis accelerometer data as a decimal value
- Serial.print(myData.yData, 2); // Prints the y-axis accelerometer data as a decimal value to two decimal places
- Serial.print(myData.zDaya, 4); // Prints the z-axis accelerometer data as a decimal value to four decimal places

SparkFun Real Time Clock (RV-8803)

The real-time clock requires the "SparkFun Qwiic RTC RV8803 Arduino Library" which configures the clock to retrieve the time and set the time down to the hundredths of a second.

Documentation: https://www.arduino.cc/reference/en/libraries/sparkfun-qwiic-rtc-rv8803-arduino-library/

Notable Structs and Functions:

RV8803 – creates a RTC object

- RV8803 rtc // Creates a new Real Time Clock object
- rtc.begin(); // Starts communication with the real time clock
- rtc.setToCompilerTime(); // Sets the real time clock to the time in the Arduino compiler
- rtc.updateTime(); // Updates the time variables in the real time clock object with the times in the RTC
- rtc.stringDateUSA(); // Return the current date in US date format: mm/dd/yyyy
- rtc.stringTime(); // Returns the current time as a string

SparkFun IoT ESP32 Microcontroller

The ESP32 microcontroller doesn't require any libraries but does require the correct configurations and downloads.

Hookup Guide: https://learn.sparkfun.com/tutorials/iot-redboard-esp32-development-board-hookup-guide/all

SD Card

Arduino has a library that has the functions needed to write to the SD card. To use the SD library, you will also need to use the SPI library.

SD.h Documentation: https://www.arduino.cc/reference/en/libraries/sd/
SPI.h Documentation:
https://www.arduino.cc/reference/en/language/functions/communication/spi/

The SPI library is bundled with every Arduino download so it doesn't need to be downloaded separately.

Notable Structs and Functions:

File – creates a file object

- File file = SD.open("/filepath.txt", FILE_APPEND); // Assigns filepath.txt from the SD card to this file object, opens this file to append new text to
- file.print("This is the first line of the file!\n"); // Adds this string of text to the file
- file.println(); // Adds another new line to the file
- file.close(); // Closes the file, cannot add any new text to this file unless you open it again

LEDs: Pinouts and Arduino

Single-Color LED

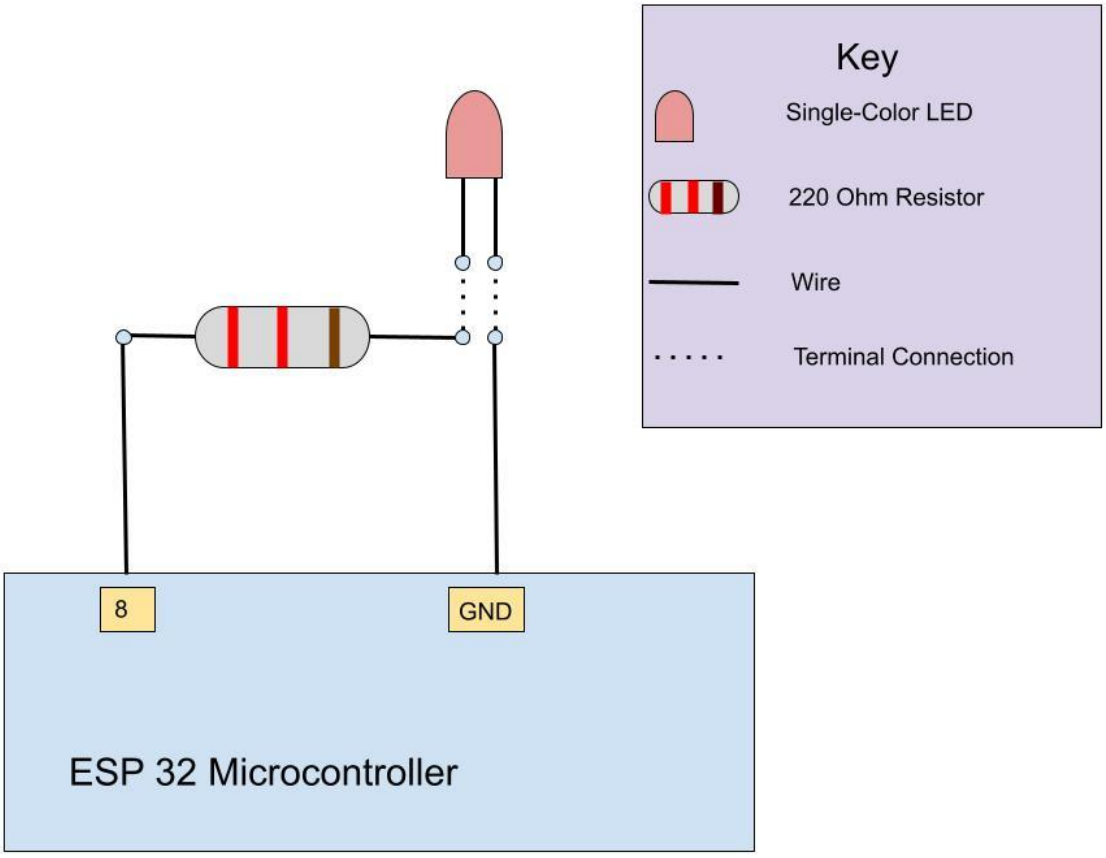

Figure 2 Breadboard Pinout for a Single-Color LED

Figure 2 shows the ESP32 microcontroller pinout an LED, which requires a 220-ohm resistor.

In Arduino:

1. Define the pin
    a. Example: #define LED-RED 8
2. In the setup() function, set the pinMode to "OUTPUT"
    a. pinMode(LED-RED, OUTPUT);
3. Set the LED to be on or off based on the condition (in the setup() or loop())
    a. ON: digitalWrite(LED-RED, HIGH);
        i. This will send 5V to that pin, which in this case is pin 8
    b. OFF: digitalWrite(LED-RED, LOW);
        i. This will cut off any power sent to pin 8, bringing the charge to 0V

Source: https://docs.arduino.cc/built-in-examples/basics/Blink

RGB LED

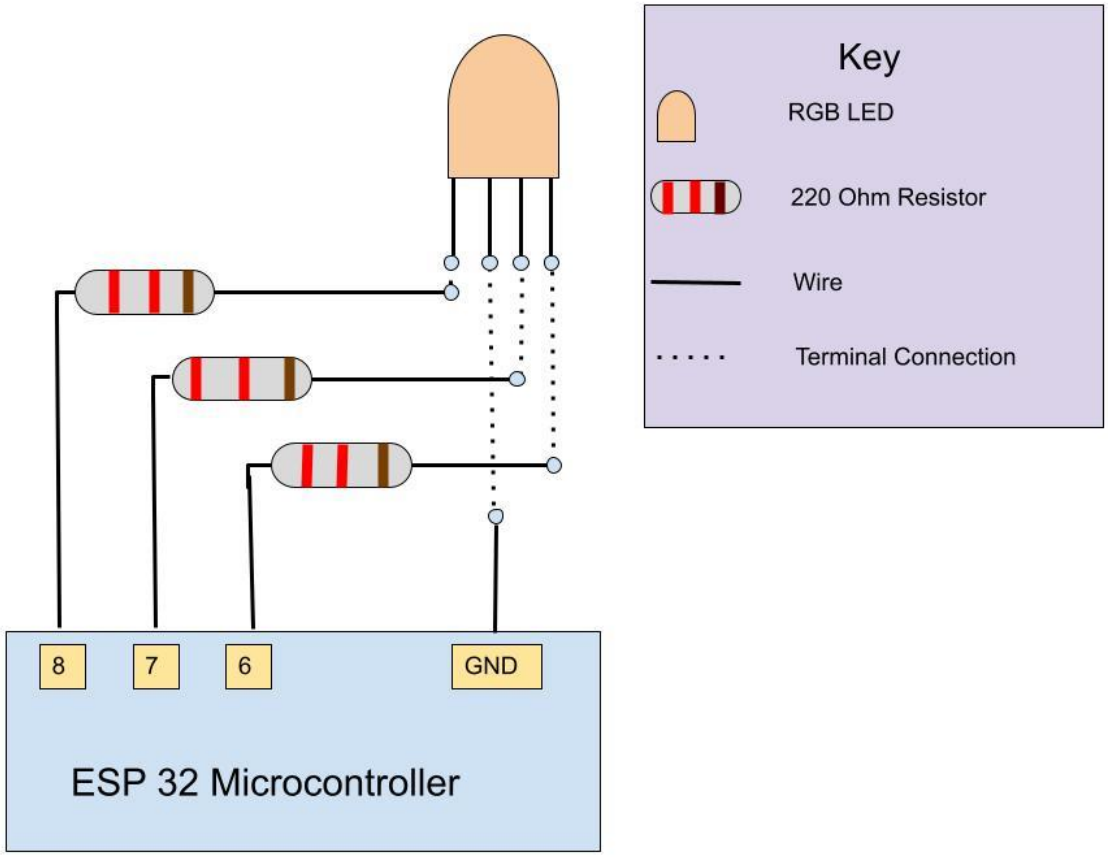

Figure 3 Breadboard Pinout for an RGB LED

Figure 3 shows the ESP32 microcontroller pinout for an RGB LED, which requires three 220 ohm resistors.

In Arduino:

1. Define the three pins for the RGB values
    a. Red Value: #define RGB-LED-1-RED 6
    b. Green Value: #define RGB-LED-2-GREEN 7
    c. Blue Value: #define RGB-LED-3-BLUE 8
2. In the setup() function, set the pinMode for each to "OUTPUT"
    a. pinMode(RGB-LED-1-RED, OUTPUT);
    b. pinMode(RGB-LED-1-GREEN, OUTPUT);
    c. pinMode(RGB-LED-1-BLUE, OUTPUT);
3. While this technically isn't required, it's likely best to create a functional called setColor that sets the specific RGB values that you want the LED to display. This can be amended based on the number of RGB LEDs by adding another variable asking for which LED to write these values to. The following example is for a program with only one RGB LED.
    a. void setColor(int redValue, int greenValue, int blueValue) {
        i. analogWrite(RGB-LED-1-RED, OUTPUT);

ii. analogWrite(RGB-LED-1-GREEN, OUTPUT);
iii. analogWrite(RGB-LED-1-BLUE, OUTPUT);

4. We can then use this function within the loop() or setup() functions to set the color of the LED
    a. setColor(255, 0, 0); // Fully red LED
    b. setColor(255, 255, 255); // White LED
    c. setColor(170, 0, 255); // Purple LED

Source: https://howtomechatronics.com/tutorials/arduino/how-to-use-a-rgb-led-with-arduino/

Bibliography

"Blink." Arduino Documentation. Accessed November 1, 2023. https://docs.arduino.cc/built-in-examples/basics/Blink.

Dejan. "How to Use a RGB Led with Arduino: Tutorial." How To Mechatronics, February 18, 2022. https://howtomechatronics.com/tutorials/arduino/how-to-use-a-rgb-led-with-arduino/.

"IOT Redboard ESP32 Development Board Hookup Guide." IoT RedBoard ESP32 Development Board Hookup Guide - SparkFun Learn. Accessed November 1, 2023. https://learn.sparkfun.com/tutorials/iot-redboard-esp32-development-board-hookup-guide/all.

"SD." SD - Arduino Reference. Accessed November 1, 2023. https://www.arduino.cc/reference/en/libraries/sd/.

"SparkFun KX13X Arduino Library." SparkFun KX13X Arduino Library - Arduino Reference. Accessed November 1, 2023. https://www.arduino.cc/reference/en/libraries/sparkfun-kx13x-arduino-library/.

"Sparkfun Qwiic RTC RV8803 Arduino Library." SparkFun Qwiic RTC RV8803 Arduino Library - Arduino Reference. Accessed November 1, 2023. https://www.arduino.cc/reference/en/libraries/sparkfun-qwiic-rtc-rv8803-arduino-library/.

"SPI." SPI - Arduino Reference. Accessed November 1, 2023. https://www.arduino.cc/reference/en/language/functions/communication/spi/.

"What Is an Open Source Library? Definition and Faqs." What is an Open Source Library? Definition and FAQs | HEAVY.AI. Accessed November 1, 2023. https://www.heavy.ai/technical-glossary/open-source-library.