Arduino Cheat Sheet

20 March 2022 09:50

Srikanth Sugavanam, 20th March 2022
ssrikanth@iitmandi.ac.in
www.srikanthsugavanam.com

Bare Minimum code needed | Arduino Documentation | Arduino Documentation

Arduino Programming Reference

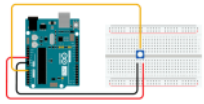
Arduino Reference - Arduino Reference

Arduino Examples

Built-in Examples | Arduino Documentation | Arduino Documentation

Analog control

Analog Read Serial | Arduino Documentation | Arduino Documentation



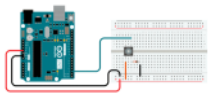
LED Dimming Example

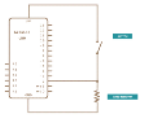
Uses PWM signals

```
/*

  Analog input, analog output, serial output

  Reads an analog input pin, maps the result to a range from 0 to 255 and uses

  the result to set the pulse width modulation (PWM) of an output pin.

  Also prints the results to the Serial Monitor.

  The circuit:

  - potentiometer connected to analog pin 0.

    Center pin of the potentiometer goes to the analog pin.

    side pins of the potentiometer go to +5V and ground

  - LED connected from digital pin 9 to ground

  created 29 Dec. 2008

  modified 9 Apr 2012

  by Tom Igoe

  This example code is in the public domain.

  http://www.arduino.cchttps://www.arduino.cc/en/Tutorial/AnalogInOutSerial

*/

// These constants won't change. They're used to give names to the pins used:

const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to

const int analogOutPin = 9; // Analog output pin that the LED is attached to

int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)

void setup() {

  // initialize serial communications at 9600 bps:

  Serial.begin(9600);
}

void loop() {

  // read the analog in value:

  sensorValue = analogRead(analogInPin);

  // map it to the range of the analog out:

  outputValue = map(sensorValue, 0, 1023, 0, 255);

  // change the analog out value:

  analogWrite(analogOutPin, outputValue);

  // print the results to the Serial Monitor:

  Serial.print("sensor = ");

  Serial.print(sensorValue);

  Serial.print("\t output = ");

  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop for the analog-to-digital

  // converter to settle after the last reading:

  delay(2);
}
```

Digital Control

How to Wire and Program a Button | Arduino Documentation | Arduino Documentation



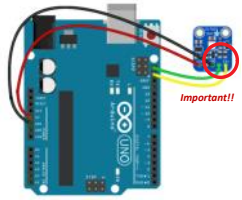


Switch debouncing

Debounce on a Pushbutton | Arduino Documentation | Arduino Documentation

```
/*
  Debounce

  Each time the input pin goes from LOW to HIGH (e.g. because of a push-button
  press), the output pin is toggled from LOW to HIGH or HIGH to LOW. There's a
  minimum delay between toggles to debounce the circuit (i.e. to ignore noise).

  The circuit:
  - LED attached from pin 13 to ground through 220 ohm resistor
  - pushbutton attached from pin 2 to +5V
  - 10 kilohm resistor attached from pin 2 to ground

  - Note: On most Arduino boards, there is already an LED on the board connected
    to pin 13, so you don't need any extra components for this example.

  created 21 Nov 2006
  by David A. Mellis
  modified 30 Aug 2011
  by Limor Fried
  modified 28 Dec 2012
  by Mike Walters
  modified 30 Aug 2016
  by Arturo Guadalupi

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Debounce
*/

// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;      // the number of the LED pin

// Variables will change:
int ledState = HIGH;         // the current state of the output pin
int buttonState;             // the current reading from the input pin
int lastButtonState = LOW;   // the previous reading from the input pin

// the following variables are unsigned longs because the time, measured in
// milliseconds, will quickly become a bigger number than can be stored in an int.
unsigned long lastDebounceTime = 0;  // the last time the output pin was toggled
unsigned long debounceDelay = 50;    // the debounce time; increase if the output flickers

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);

  // set initial LED state
  digitalWrite(ledPin, ledState);
}

void loop() {
  // read the state of the switch into a local variable:
  int reading = digitalRead(buttonPin);

  // check to see if you just pressed the button
  // (i.e. the input went from LOW to HIGH), and you've waited long enough
  // since the last press to ignore any noise:

  // If the switch changed, due to noise or pressing:
  if (reading != lastButtonState) {
    // reset the debouncing timer
    lastDebounceTime = millis();
  }

  if ((millis() - lastDebounceTime) > debounceDelay) {
    // whatever the reading is at, it's been there for longer than the debounce
    // delay, so take it as the actual current state:

    // if the button state has changed:
    if (reading != buttonState) {
      buttonState = reading;

      // only toggle the LED if the new button state is HIGH
      if (buttonState == HIGH) {
        ledState = !ledState;
      }
    }
  }

  // set the LED:
  digitalWrite(ledPin, ledState);

  // save the reading. Next time through the loop, it'll be the lastButtonState:
  lastButtonState = reading;
}
```

Sensors and Communications protocols



Install Adafruit_Si7021 library

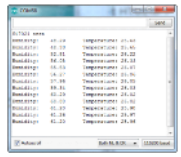
To begin reading sensor data, you will need to install the Adafruit_Si7021 library (code on our github repository). It is available from the Arduino library manager so we recommend using that.

From the IDE open up the library manager...

And type in **adafruit si7021** to locate the library. Click **install**

Also install the **Adafruit Unified Sensor** library in the same way:

Arduino Code | Adafruit Si7021 Temperature + Humidity Sensor | Adafruit Learning System



Moving from Arduino to breadboard

From Arduino to a Microcontroller on a Breadboard | Arduino Documentation | Arduino Documentation