

---

## Arduino Project

# SmartStart Interval Timer

## OVERVIEW

The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.

## GOALS

1. Set up an Arduino UNO and components in such a way that it can be used as a robust interval timer
2. Create and upload the software needed to make everything work as expected.

## SPECIFICATIONS

Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

For writing and uploading software, one can use the convenience of the ARDUINO IDE.

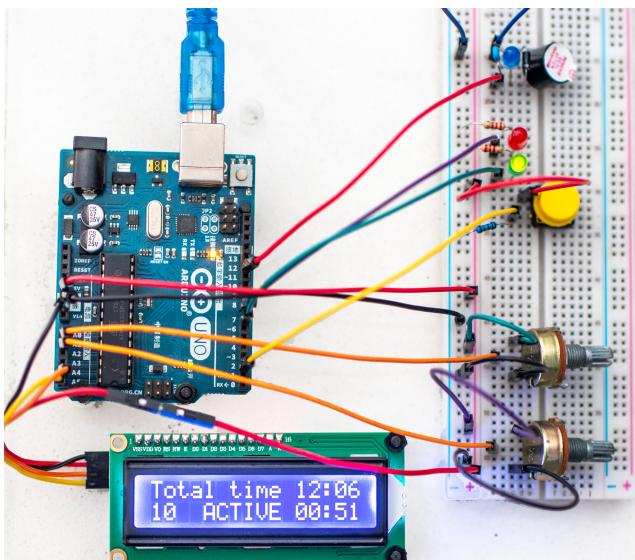
<https://docs.arduino.cc/software/ide-v1>

## MILESTONES

### Hardware

Well, first of all, one needs to get all the hardware needed, but fortunately that's relatively inexpensive and readily available from places like [Micro Rotics](#) (see component list)

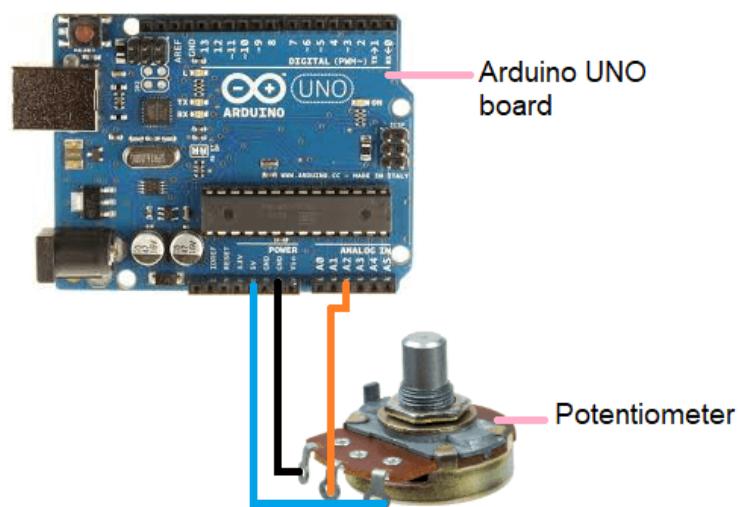
Firstly, build everything out on a breadboard in order to test and adjust as needed.



### Potentiometers

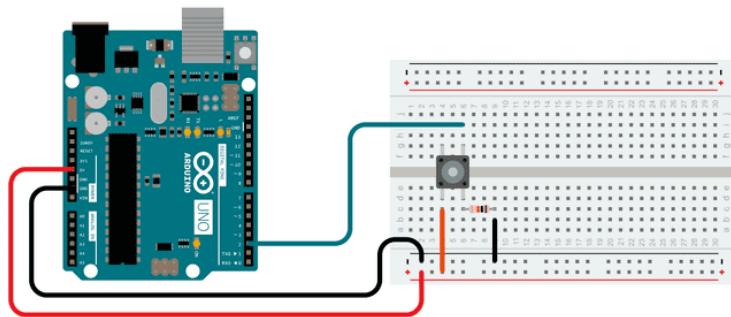
These are quite straightforward. Potentiometers are variable resistors with a center pin; in this configuration they also serve as voltage divider that, when interpreted as a ratio, can provide a range between 0% and 100%. When read digitally it will be a value between 0 and 1023.

To connect them, attach one outside pin to the 5V, and another outside pin to GND. The center pin then gets connected to one of the Arduino Analog Inputs. In this case there are two Pots, connected to A0 and A1 respectively.



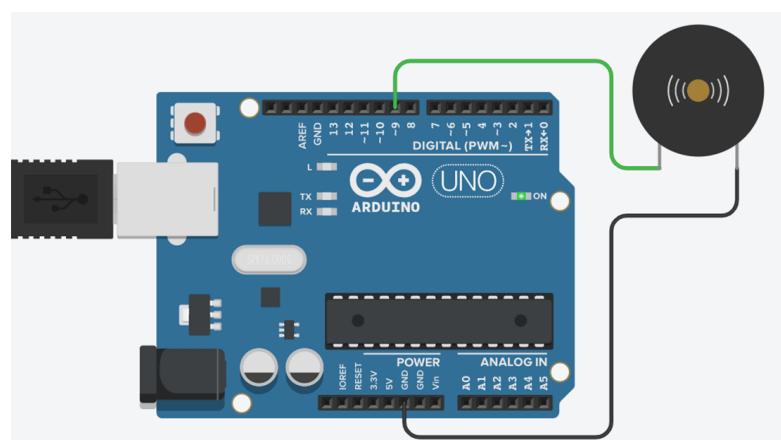
### Push Button

Of course we need a push button to start and stop the operation. Here we basically just create an open circuit from 5V to a digital pin ( 2 ) on the board. One thing to note is that one needs to add a high value resistor ( 10k ohm) to ground, otherwise we get what's called floating voltage on the Digital input, which leads to regular mis reads.



## Buzzer

The buzzer is a significant feature in this project, but we must take care because it's an inductive device, similar to a DC motor, and should preferably use an outside power supply with a trigger like NPN Type transistor. But since this is a small load there are some measures we can put in place. Connect the + side of the buzzer to a digital output pin ( 12 ) and the other side to GND.



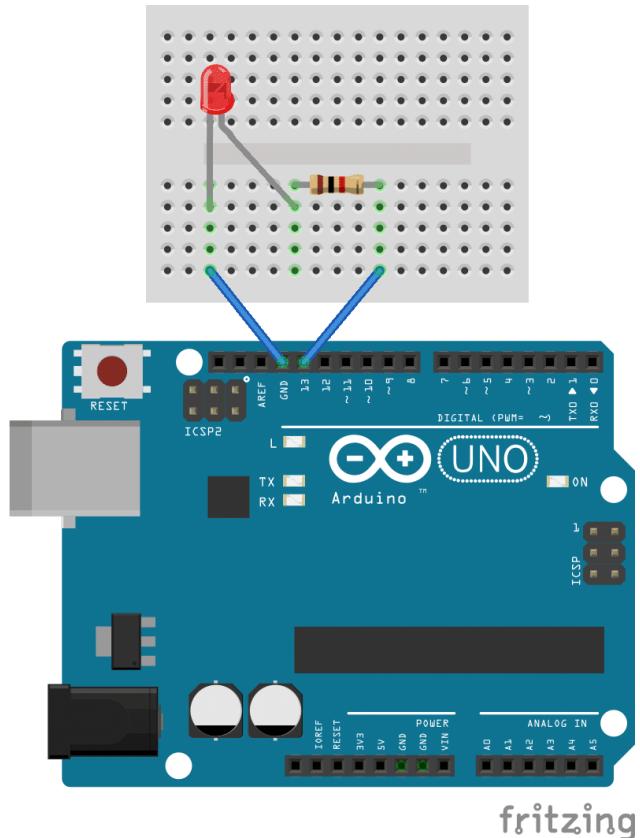
Now, add a diode in parallel to the buzzer, but in the reverse direction. In other words, the cathode (white strip) on the 5V side. Also add a small capacitor in parallel. One can also use an LED in parallel, then neither buzzer nor LED needs a resistor, but if you're more comfortable a small resistor can be added in series.

## LEDs

Of course LEDs are a good visual indicator, so we'll use Green for the Active cycle and red for the Rest cycle in this example. LEDs operate at slightly different voltages, but all below the 5V supply so they need a resistor in series. For perfect brightness balance it would be something like 2K ohm for the red, and 1K2 for the green LED, but it is ok to just use 2K resistors.

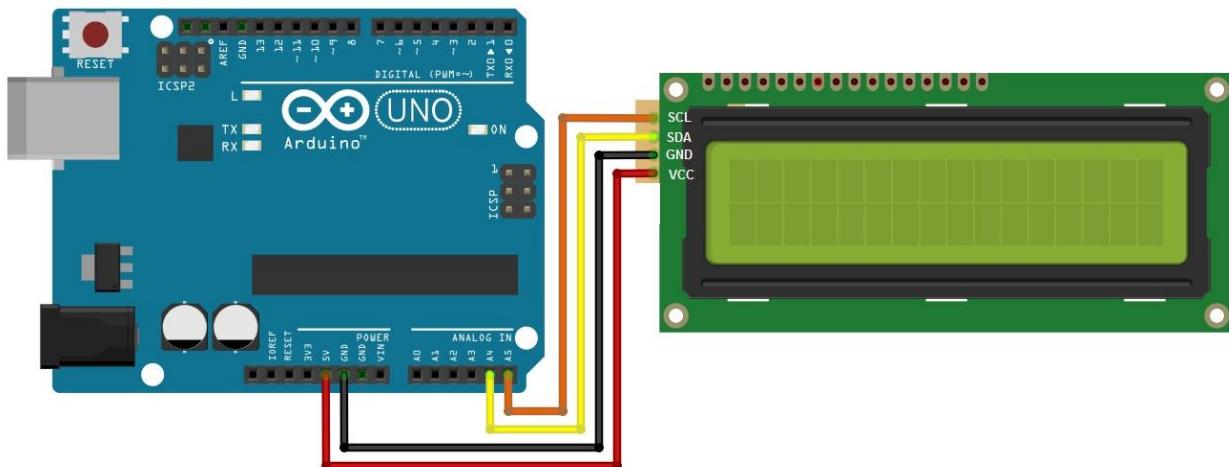
Prototyping resistors also serve as an easy way to link components to pin, so one could opt to use 1K resistor on the 5V(pin) side and 1K on the GND side.

Go ahead and connect the resistor and LED in a series circuit. In this case Green to the pin 7 and red to pin 8. Keep in mind that the cathode (short leg) must be on the GND line, and the long leg on the pin side.



## LCD

Of course the LCD comes in very handy. Here we use the I<sup>2</sup>C functionality (software aided) to only use 4 wires V<sub>S</sub>, GND, SDA (data), and SCL (clock). See diagram.



---

## Software

Assuming that you've already installed the Arduino IDE, go ahead and copy the project code (main.cpp) to your project's main. The code was optimized for easy reading, and not simplicity or performance. But we're not trying to put a man on the moon with this, so it really doesn't matter all that much.

Very important here is that one also adds a library to use with the LCDdisplay. Especially in this case where we use the 4-wire I2C type interpreter.

The code consists of Declarations and initial setup, as well as a main Loop, and functions. The main loop does exactly as the name implies, it loops through the code for execution, and when all is done, starts back at the loop. The functions are code blocks that get executed when called from within the Loop, but they must be declared before.

The interesting bit here is the use of a time-stack, which breaks the loop down into sections depending on time interval, so that we don't execute all the code with every loop. In this case it's not particularly clear, but one could therefore limit code execution, like updating the LCD to every other loop.

Unfortunately I don't have too much time to go through the software as much as one would like, but there are ample serial readout lines which are commented out. So if you see something like // Serial.print, then you can uncomment it by removing the // in the beginning, and you'll be able to see feedback from the Arduino in the IDEs console. Just keep in mind that the chip has limited capacity, and if you uncomment all the Serial.print lines, it will cause either upload, or device program execution to fail.

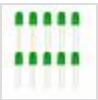
Once you've familiarized yourself with the code, go ahead and upload it to the Arduino, via the upload utility. Make sure that you use the correct board choice, in this case Arduino UNO R3.

---

## Component list

	1	UNO R3 - Compatible with Arduino®	UNOR3	R219.00
	1	Uno Protoshield	UNO-PROTO	R29.00
	1	USB AB Cable 50cm	USBAB50	R15.00
	4	Header Female 10 Pin 2.54mm (10 Pack)	HFST-10-TH254	R15.00
	1	Resistor 10K (50 Pack) ***	RES-10K-025	R14.00
	2	10K Potentiometer Linear (4 Pack)	10K-POT	R22.00
	2	Knob for Potentiometer Black with Blue Insert	WH148-BLUE	R7.00
	1	Buzzer 5V Low Profile PCB (4 Pack)	BUZ-50V-PCB	R16.00

---

	1	Capacitor Kit	CAPKIT	R18.00
	1	LED Blue 5MM - 17mm (10 Pack)	LED-05-BLU-17	R10.00
	1	LED Red 5MM (10 Pack)	LED-RED-5MM	R10.00
	1	LED Green 5MM - 17mm (10 Pack)	LED-05-GRN-17	R10.00
	4	Resistor 1K (50 Pack)	RES-1K-50	R14.00
	1	LCD 16x2 Character Display, White on Blue Background, I2C Interface, 5V	LCD1602-WB-5V-I2C	R68.00
	1	Diode 1N4007 (10 Pack)	1N4007-10	R15.00
	1	Switch, Tactile Round Button - 7 Colours	TACTILE7	R18.00