Arduino Hands-On: Presence Detection

Alberto Perro - CERN - 8/03/2023

GitHub: @albydnc
CERN: alberto.perro

Who I am

I’m Alberto Perro.

I am a PhD student in Physics and I work at LHCb.

I’ve been using Arduino since 2012 and I’ve been teaching it since 2014.

I also used Arduino in my past work for industrial and commercial projects.

Now I’m here to teach you more about it :)

Outline

- Description of the project
- Hardware
    - Arduino
    - Addressable LEDs
    - PIR sensor
- Software
    - Arduino IDE
- Libraries
    - FastLED
- Hands-on Lab

Presence detector

This project is a simple introduction to the world of Arduino.

We will see what is an Arduino, how to program it, and how to interact with the environment.

In this workshop you will learn how to build a presence detector system that will alert you if there is somebody near.

PIR Sensor: How does it work?

A passive infrared sensor (PIR sensor) is an electronic sensor that detects infrared light radiating from objects.

Presence of a person is determined by the change of average temperature from the room temperature to the body temperature.

Presence Detector: Project sketch

Our Arduino will be programmed in a simplified C++ language, often called Arduino language.

In our program, we will gather the presence information and we will make a luminous alert with the addressable LEDs.

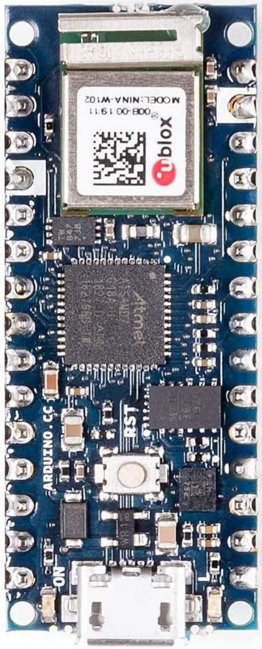
Hardware

Arduino Nano 33 IoT

This is the brain of our project and you'll learn how to program it.

Its key features:

- ARM Microcontroller @ 48 MHz
- 14 Digital I/O @ 3.3V
- 5 Analog Inputs
- Built-In WiFi and Bluetooth

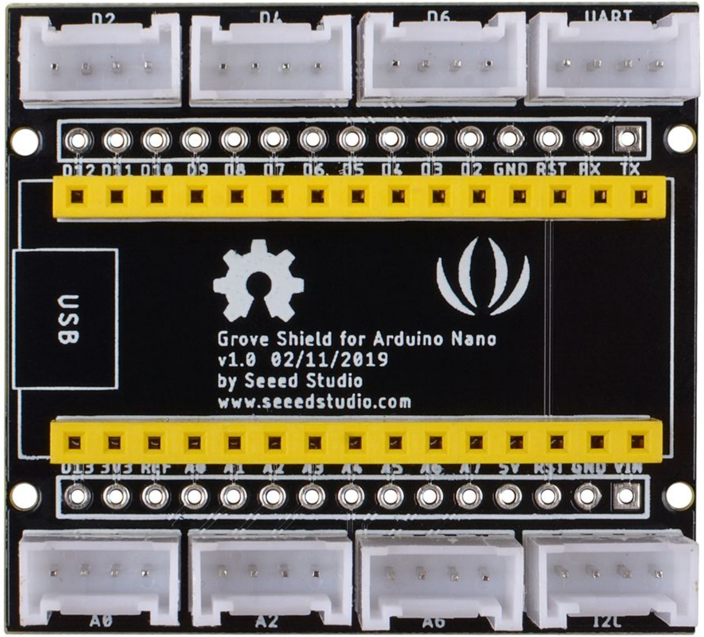


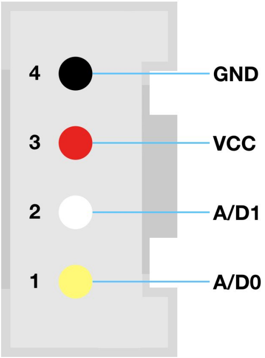
Grove Nano Shield

The Grove Nano Shield breaks out the pins on the Arduino into Grove connectors.

The Grove connector is a standard connector that simplifies prototyping.

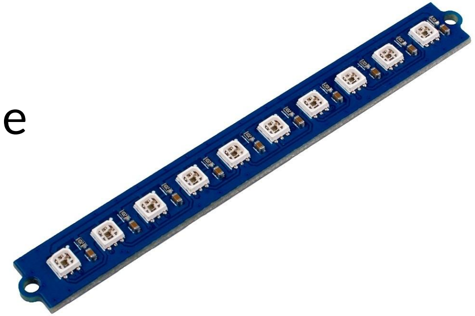
Grove carries power (voltage is selectable on the shield) and two signal wires.


D2
D4
D6
UART
D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX TX
USB
Grove Shield for Arduino Nano
v1.0 02/11/2019
by Seeed Studio
www.seeedstudio.com
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN
A0
A2
A6
I2C



Addressable LEDs

These are the LEDs we will use as a *Human-Machine Interface.*



Addressable means that we can decide each LED's color and brightness.

Key features:

- 10 Addressable RGB LEDs

PIR Motion Sensor

This sensor measures IR radiation incoming.

It has a 100 degree detection angle and a response time of under 1 second.

Its output is digital, active high.

Software

Arduino IDE

The arduino IDE is a free and open-source development environment.

It enables programming and debugging Arduino boards and third-party boards as well.

It gives access to many open-source libraries which can be integrated in your code.

We will use this tool to write our project.

Libraries

FastLED

Since our LEDs are addressable, it is not as easy as flipping a pin ON and OFF.

Addressable LEDs use a protocol that carries the RGB channels' data for each LED in the strip.

To encode the data and drive the LEDs, we will use the `FastLED` library.

Hands-On Lab

Blink

This example is used to test your board and understand the code structure of an Arduino program.

Each program is divided in two section: `setup()` and `loop()`.

In `setup()` we will put the code that will be run at the beginning.

The code written in `loop()` is repeated infinitely.

Blink

On the board, there is a LED connected to pin 13.

First, we need to tell the Arduino where the LED is and that it is an output. This is done with the `pinMode(pin, mode)` command and must be put in the setup.

Second, we need to drive the led pin using the `digitalWrite(pin, value)` command. We also need some `delay(ms)` to make the blink visible to the human eye.

Addressable Blink

The FastLED library comes in help when using addressable LEDs.

We need a global array to store the LED colors (`CRGB leds[NUM_LEDS]`) and then we need to setup the FastLED library by telling it what LEDs we have, how many of them we have, and on which pin they are connected.

```
FastLED.addLeds<NEOPIXEL,DATA_PIN>(leds,NUM_LEDS)
```

After that, we can assign values to the array elements and display the colors with `FastLED.show()`.

Reading the PIR Sensor

To interact with the sensor, we can use the built-in arduino library.

In `setup()`, we'll need to initialize the pin by doing `pinMode(PIR_PIN,INPUT)`.

In the `loop()`, we read the measurements by using `digitalRead(PIR_PIN)`.

The interval between measurements should be at least 1 second.

Printing the values

To see the readings, we can use the `Serial` library: by putting `Serial.begin(baud)` in the setup and `Serial.println(str)` in the code, we can print the data in the serial monitor of the IDE.

Using the Serial Plotter Tool

We can also graph the readings by using the Serial Plotter Tool available in the Arduino IDE.

We simply need to print the readings following the specification:

```
Variable1:VALUE,Variable2:VALUE
```

Project Code and Documentation

https://gitlab.cern.ch/alperro/arduino-presence-detector

References

https://www.arduino.cc/en/software

https://fastled.io/

https://wiki.seeedstudio.com/Grove-Digital-PIR-Sensor