TOPIC

C# on CHIPs

.NET on MCU

Mirco Vanini
Microsoft MVP Developer Technologies

Thanks to our partners

About me

Mirco Vanini

Microsoft MVP Developer Technologies

Consultant focused on industrial and embedded solutions using .NET and other native SDKs with over 30 years of experience, XeDotNet community co-founder, speaker and Microsoft MVP since 2012

@MircoVanini
www.proxsoft.it
https://www.linkedin.com/in/proxsoft

#CodeGen2023

@cloudgen_verona

Agenda

- .NET for IoT
- MCU, CPU, what's the difference?
- .NET Micro Framework
- Alternative to .NET MF
- What is nano framework
- Demo

.NET for IoT

- Full integration with Azure services (IoT Hub, IoT Edge)
- ARM32 and ARM64 Runtime and SDK build:
  - Ubuntu 16.04+, Debian 9+, Alpine 3.9+
  - Windows 1809+
  - Docker images
- API for all the major IoT Device protocols: GPIO, PWM, SPI, I2C, …
- Device binding for a growing set of peripherals

.NET IoT Libraries

MCU, CPU, what's the difference?

- A Micro Controller Unit (MCU) is a small processor, usually a synonym for low power consumption, and small size. Modern ones like STM32F7, ESP32, TI CC1352R are ARM Cortex M based, embedding few hundreds of kilobytes of RAM, small flash and a decent few hundred million of hertz clock cadence.

- A Central Processing Unit (CPU) is a larger processor, in this category, you'll find all the traditional x86, x64, but as well other ARM processors like the one used in Raspberry Pi.

.NET Micro Framework

The .NET Micro Framework (NETMF) is a .NET Framework platform for resource-constrained devices with at least 512 kB of flash and 256 kB of random-access memory (RAM).

It includes a small version of the .NET Common Language Runtime (CLR) and supports development in C#, Visual Basic .NET, and debugging (in an emulator or on hardware) using Microsoft Visual Studio.

.NET Micro Framework (2002->2015)

In 2002 at Comdex Microsoft annunced Smart Personal Objects Technology, the first devices to make use of SPOT were released in 2004 by Fossil and Suunto

In November 2009, Microsoft released the source code of the Micro Framework to the development community as free and open-source software

In January 2010, Microsoft launched the netmf.com community development site to coordinate ongoing development of the core implementation with the open-source community

.NET Micro Framework (2002->2015)

In 2010 Microsoft released the The Gadgeteer platform, an open-source rapid-prototyping standard for building small electronic devices using the Microsoft .NET Micro Framework

On 9 January 2010, GHI Electronics announced FEZ Domino

On 3 August 2010, Secret Labs announced the Netduino

In 2015 Microsoft released v4.4, the last release of Micro Framework

.NET Micro Framework (2002->2015)

On 16 December 2016, GHI Electronics announced their own implementation of Micro Framework called TinyCLR OS

On 23 January 2017 a group of embedded systems developers publicly announced .NET nanoFramework as spin-off of .NET Micro Framework.

Alternative to .NET MF

nano FRAMEWORK

TinyCLR OS

meadow

What is nano framework

.NET nanoFramework is a free and open-source platform that enables the writing of managed code applications for constrained embedded devices.

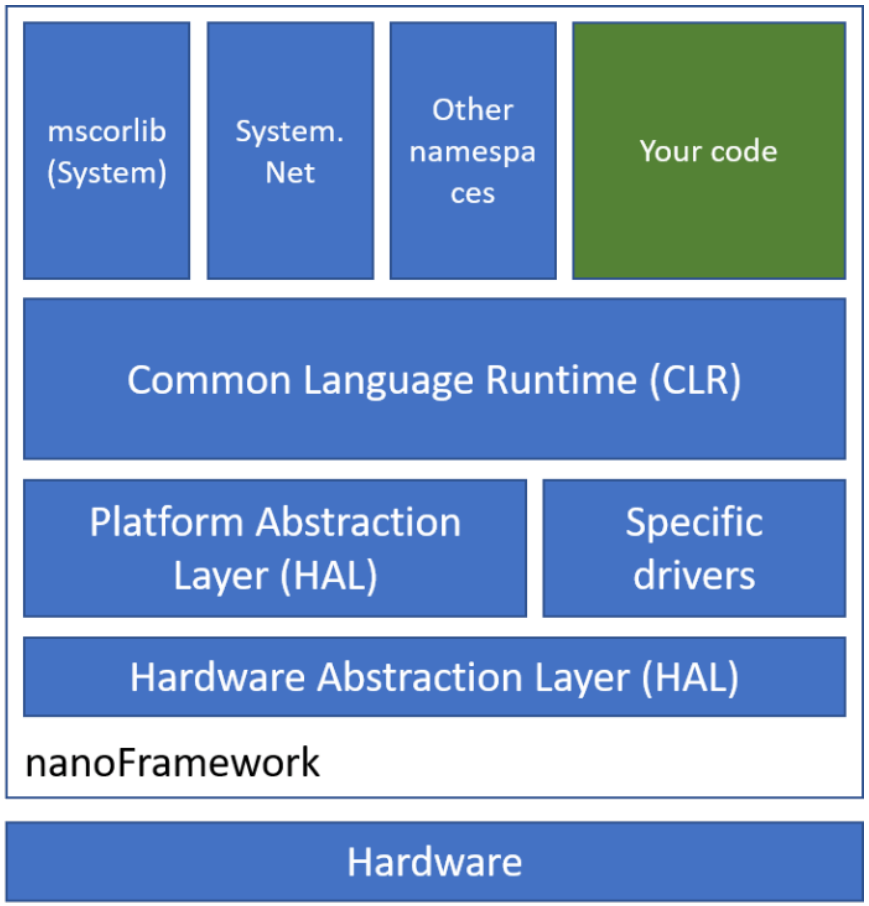
It is suitable for many types of projects including IoT sensors, wearables, academic proof of concept, robotics, hobbyist/makers creations or even complex industrial equipment.

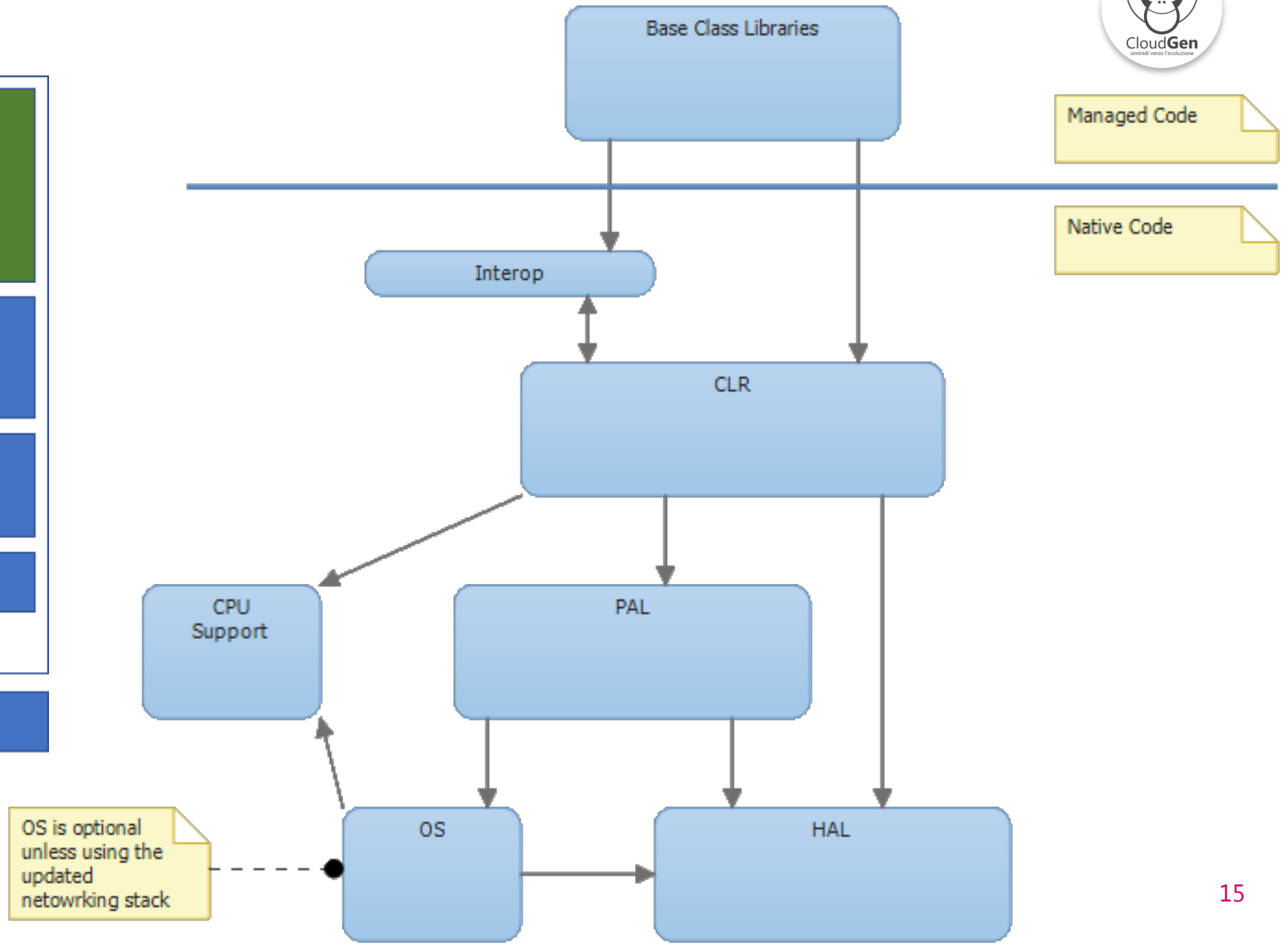
It includes a reduced version of the .NET Common Language Runtime (CLR) and features a subset of the .NET base class libraries along with the most common APIs included in .NET IoT allowing code reuse from .NET IoT applications, thousands of code examples and open source projects.

What is nano framework

- Can run on resource-constrained devices with as low as 128kB of flash and 64kB of RAM.
- Runs directly on bare metal. Currently ARM Cortex-M and ESP32 devices are supported.
- Supports common embedded peripherals and interconnects like GPIO, UART, SPI, I2C, USB, networking.
- Provides multi threading support natively.
- Support for energy-efficient operation such as devices running on batteries.
- Support for Interop code allowing developers to easily write libraries that have both managed (C#) and native code (C/C++).
- No manual memory management because of its simpler mark-and-sweep garbage collector.
- Execution constrains to catch device lockups and crashes.

Layers


mscorlib (System)
System. Net
Other namespa ces
Your code
Common Language Runtime (CLR)
Platform Abstraction Layer (HAL)
Specific drivers
Hardware Abstraction Layer (HAL)
nanoFramework
Hardware

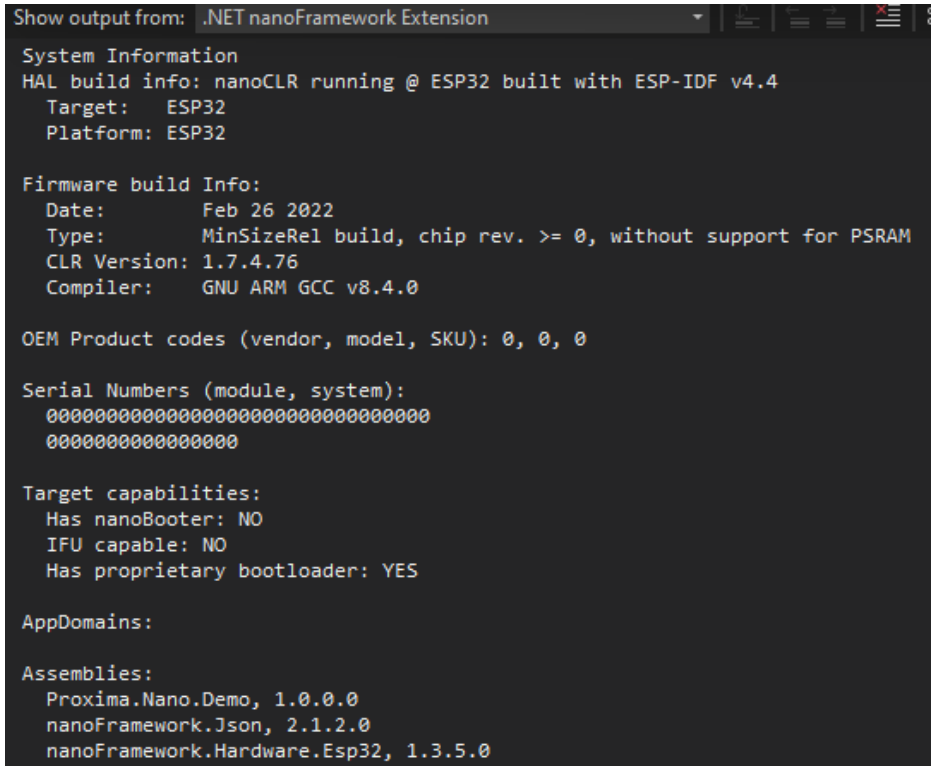

Base Class Libraries
Managed Code
Native Code
Interop
CLR
CPU Support
PAL
OS is optional unless using the updated netowrking stack
OS
HAL

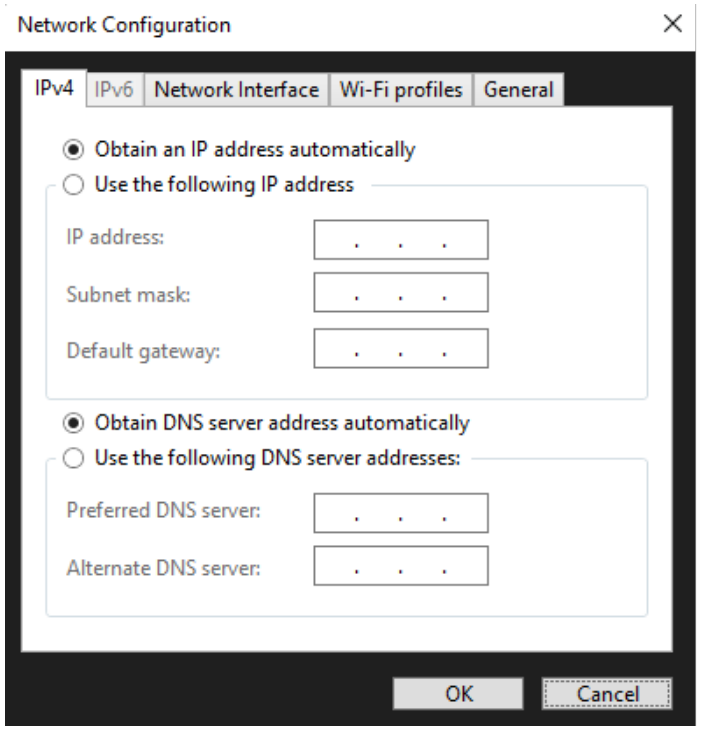
nano framework - getting started

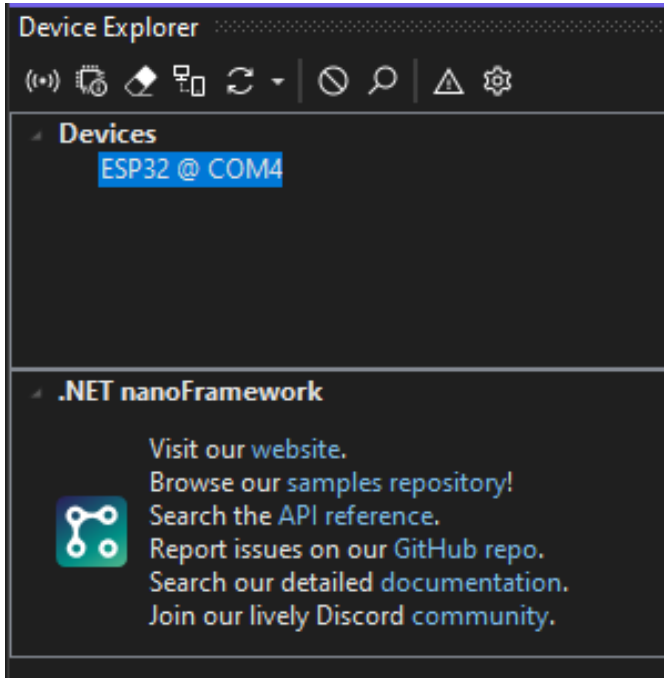
- Install Visual Studio, make sure to install the workloads for .NET desktop development and .NET Core cross-platform development.
- Install the nanoFramework extension for Visual Studio
- Install nano Firmware Flasher (nanoff) tool, the .NET 6.0 Runtime (or .NET 6.0 SDK) must be installed
- Uploading the firmware to the board using nanoFirmwareFlasher
- Run Visual studio, File > New > Project menu, to open the Create a new project, Enter nanoFramework into the Search for templates search prompt. Choose the Blank Application (nanoFramework) template and press the Next button

.NET nanoFramework Home
nanoframework.docs
nanoFramework.IoT.Device

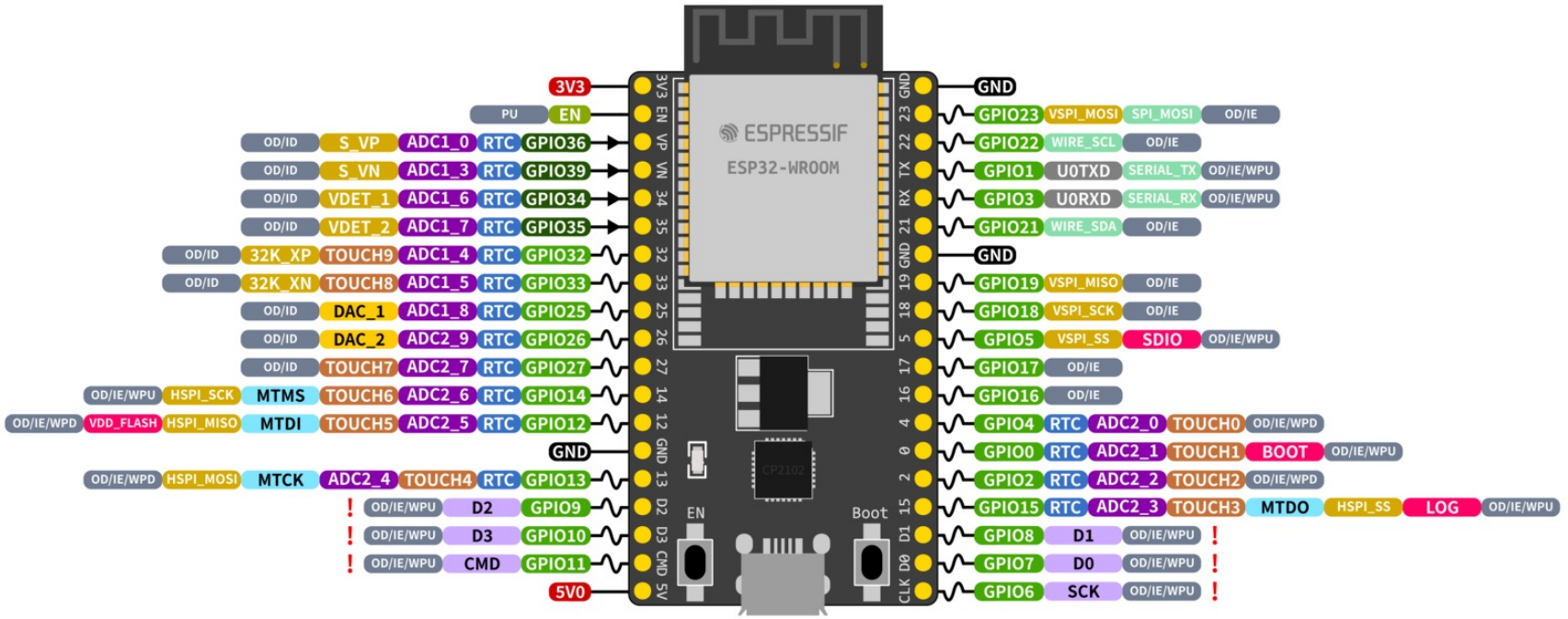
nano framework – VS2022 integration







ESP32-DevKitC



ESP32 Specs

32-bit Xtensa® dual-core @240MHz
Wi-Fi IEEE 802.11 b/g/n 2.4GHz
BLuetooth 4.2 BR/EDR and BLE
520 KB SRAM (16 KB for cache)
448 KB ROM
34 GPIOs, 4x SPI, 3x UART, 2x I2C,
2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,
1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet

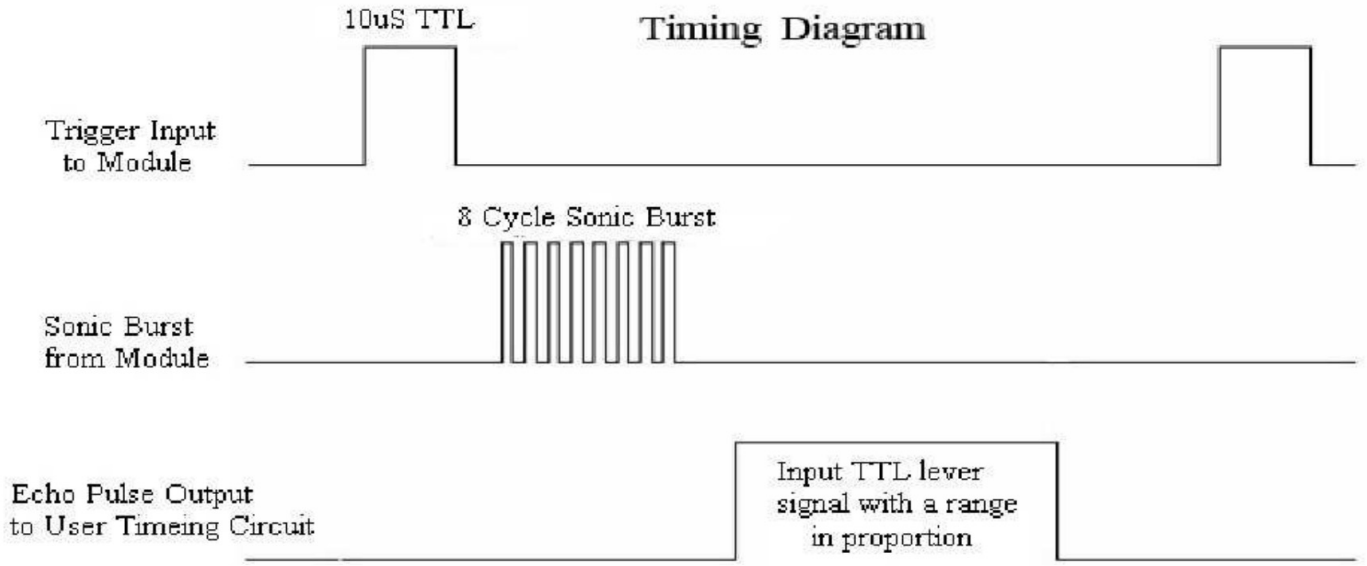
PWM Capable Pin
GPIOX GPIO Input Only
GPIOX GPIO Input and Output
DAC_X Digital-to-Analog Converter
DEBUG JTAG for Debugging
FLASH External Flash Memory (SPI)
ADCX_CH Analog-to-Digital Converter
TOUCHX Touch Sensor Input Channel
OTHER Other Related Functions
SERIAL Serial for Debug/Programming
ARDUINO Arduino Related Functions
STRAP Strapping Pin Functions

RTC RTC Power Domain (VDD3P3_RTC)
GND Ground
PWD Power Rails (3V3 and 5V)
! Pin Shared with the Flash Memory *Can't be used as regular GPIO*

GPIO STATE
WPU: *Weak Pull-up (Internal)*
WPD: *Weak Pull-down (Internal)*
PU: *Pull-up (External)*
IE: *Input Enable (After Reset)*
ID: *Input Disabled (After Reset)*
OE: *Output Enable (After Reset)*
OD: *Output Disabled (After Reset)*

Ultrasonic Ranging Module HC - SR04

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.

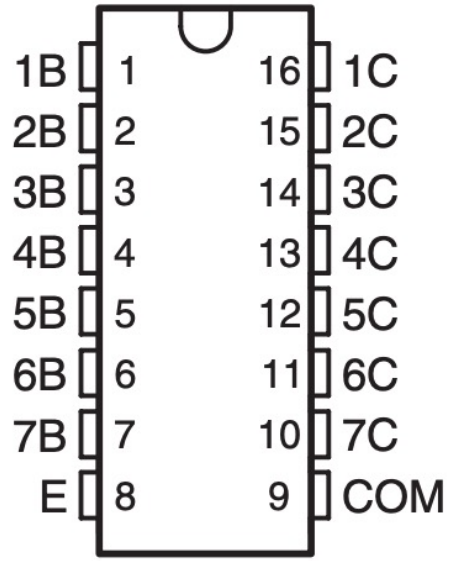


Stepper Motor 5V 4-Phase 5-Wire & ULN2003 Driver Board

FEATURES

- 500-mA-Rated Collector Current (Single Output)
- High-Voltage Outputs: 50 V
- Output Clamp Diodes
- Inputs Compatible With Various Types of Logic
- Relay-Driver Applications

ULN2002A . . . N PACKAGE
ULN2003A . . . D, N, NS, OR PW PACKAGE
ULN2004A . . . D, N, OR NS PACKAGE
ULQ2003A, ULQ2004A . . . D OR N PACKAGE
(TOP VIEW)



Demo

Successfully invoked method 'GetDoorStatus' on device 'proxima-nano-demo' with response {"status":200,"payload":{"doorStatus":"close"}}.

2:22:21 PM

Telemetry

Consumer group: $Default

Specify enqueue time: No

Use built-in event hub: Yes

Receiving events...

Thu Mar 17 2022 14:15:29 GMT+0100 (Central European Standard Time):


```
{
  "body": {
    "EventType": 2,
    "Name": "mainDoor",
    "Sender": "proxima-nano-demo",
    "DataTicks": 637831197298779600,
    "Id": "bdfa23e7-cb31-495b-e789-8eb8550e0e30"
  },
  "enqueuedTime": "Thu Mar 17 2022 14:15:29 GMT+0100
  "properties": {}
}
```

Thank you

Any questions?


https://github.com/MircoVanini


@MircoVanini


https://www.linkedin.com/in/proxsoft/

Microsoft MVP Developer Technologies