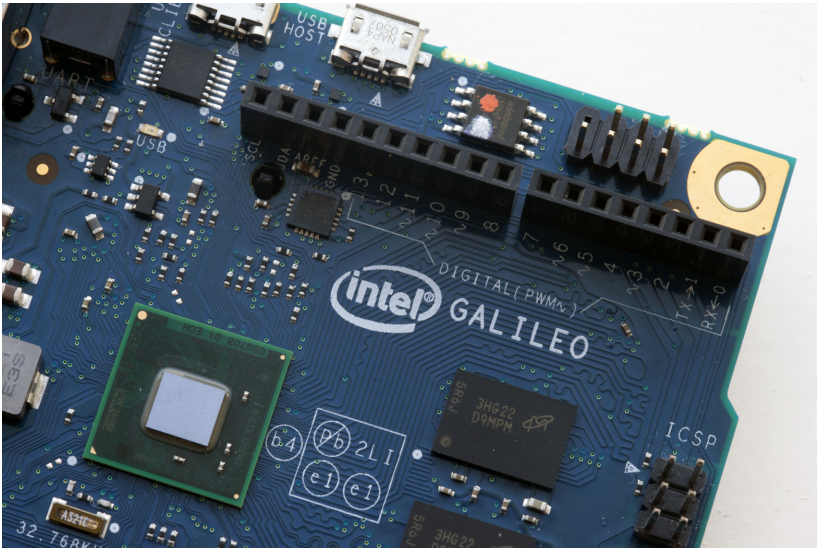


1/Introduction to Galileo



The purpose of the hardware and software that make up the Arduino platform is to reduce complexity when making an electronic project. It's meant to let you experiment, invent, and explore creative uses of technology rather than getting bogged down in technical mastery. By offering compatibility with Arduino hardware and software, Intel Galileo delivers an easy-to-use platform but has more power and features than typical Arduino boards.

What Is Galileo?

Galileo is a *hardware development board*, which is an electronic circuit board that helps you develop interactive objects by reading information from the physical world, processing it, and then taking action in the physical world. If it's connected to a network, it can also communicate to other devices like web servers. Galileo is an Arduino-compatible development board.

What Is Arduino?

There are a few answers to the question, "What is Arduino?" First and foremost, it's a hardware development board like Intel's Galileo. There are models of boards such as the Arduino Uno, Arduino Mega, and Arduino Yún. Each of these Arduino boards has different capabilities. The most basic board, the Arduino Uno, is typically what people are referring to when they say "an Arduino."

There's also the Arduino IDE software, which is the computer application that you use write code and upload it to the board. Arduino is also the name of the language used to program the board.

If you're entirely unfamiliar with Arduino and want to learn more about it, [the Arduino website](#) has many resources including getting started guides, reference information, communities, projects, and news updates. The book [Getting Started with Arduino](#) by Massimo Banzi (O'Reilly) was my first guide to using the popular development board. It covers the design philosophy of Arduino ("The Arduino Way") and walks you through the basics of using it. This book will cover a lot of that ground as well, but tailored for the Galileo board.

Galileo is an *Arduino-compatible* board, meaning that it can be programmed with the Arduino IDE using the Arduino programming language. It's also compatible with the Arduino 1.0 pinout, the design specification that says which pins go where on the board. Because it's compatible with the Arduino 1.0 pinout, you're able to attach most *Arduino shields*. A shield sits on top of the board and expands the functionality of it. Common circuits to drive motors, control many LEDs, or play sounds can come in the form of shields. The pin layout compatibility also makes it easy to use Galileo when you're following tutorials written for the other Arduino boards.

Inputs and Outputs

Like other hardware development boards, Galileo reads inputs and can control outputs. An *input* brings information from the physical world into the board's processor. It can be as simple as the state of a button or switch but can also be the position of a dial or slider like you see on a sound mixing board. Sensors can also be used as inputs (see [Figure 1-1](#)) to read information from

the physical world. There are plenty of sensors to choose from including temperature, light level, sound level, acceleration, and much more.

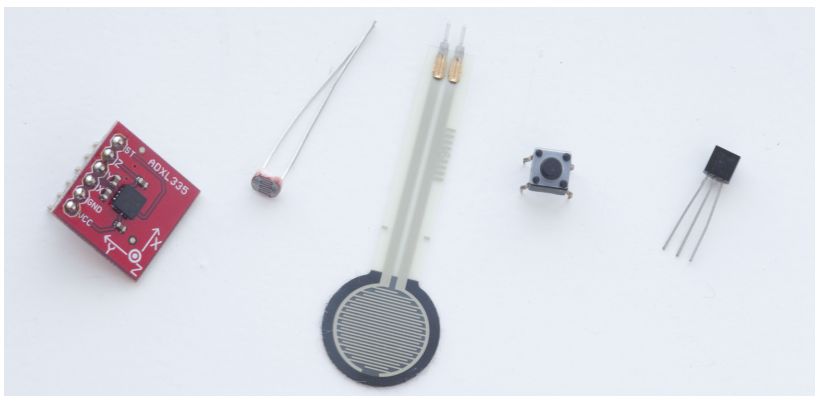


Figure 1-1. A few possible inputs. From left to right: an accelerometer, a photo cell, a pressure sensor, a button, and a temperature sensor.

An *output* is how a development board like the Galileo can affect the physical world. It can be as simple as a *light emitting diode*, or LED, which glows when electrical current runs through it. An LED might indicate whether the device is turned on, or if there's an error (a blinking red LED would be perfect for that). Outputs could also be motors that drive wheels on a robot, a text display for the temperature, or a speaker that plays musical tones. [Figure 1-2](#) shows a few.

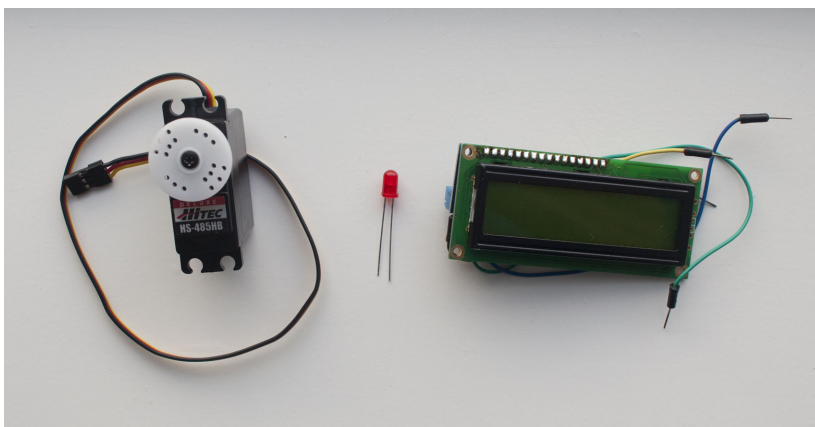


Figure 1-2. A few possible outputs. From left to right: a servo motor, a light emitting diode, and an LCD character display.

For example, a simple stopwatch has inputs and outputs. The start button would be considered an input. When you press the start button, it triggers a timer that keeps track of the time and outputs that information to the display on the face of the watch.

A digital voice recorder has a microphone for sound input, and a small speaker for sound output. Like the stopwatch, it also has input buttons to start or stop the recorder and a small display to output the amount of time that's left to record before you fill up the device's memory.

Code

Of course, it's not as simple as just wiring up inputs and outputs to a Galileo. You have to tell the board how you want it to respond to the inputs and how you want it to control the outputs. By programming the board, you'll be able to tell it what you want it to do.

For instance, a simple thermostat project will periodically check the value from a temperature sensor and compare it to the desired temperature that the user set using a dial control. If the temperature that the sensor measures is lower than the desired temperature, the board will activate a heater until the temperature gets close enough to the desired temperature. Logic like this will be defined by the code you write.

The Galileo can be programmed and reprogrammed over and over again. In fact, in the course of developing a project, you'll likely go through a cycle of writing code, uploading it to the board, checking how it works, finding problems, making adjustments to your code, and then uploading it again.

You may even find yourself using the board for one project, and then pulling the board out, reprogramming it, and using it for a completely different project a few weeks later.

Communication

The Galileo can also communicate with other devices in a few different ways. You can have it connect to your computer via USB to send and receive data. You might have Galileo send information about what it's doing to a console window running on your computer so that you can figure out why something isn't working right (this is known as *debugging*). Or you can have it send information about sensors to the computer so that it can display a live graph.

Galileo can also connect to other devices over the Internet using its built-in Ethernet ([Figure 1-3](#)) or an optional WiFi module. It can receive information about the weather or your email. It can search Twitter and much more. It can also use the Internet connection to send information such as temperature sensor data, the images from a webcam, or the state of your dog's water bowl.

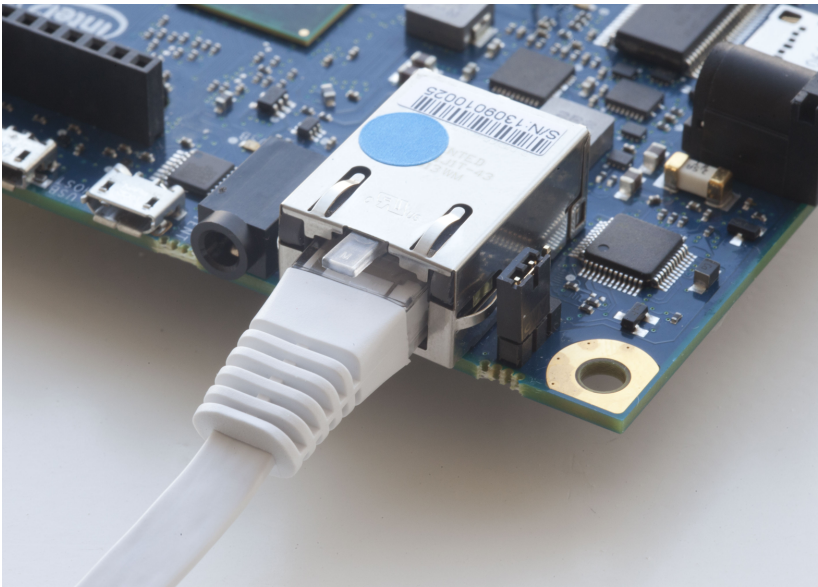


Figure 1-3. *The Galileo's Ethernet port is just one way it can communicate with users or other devices.*

What Makes Galileo Different?

If you've used a typical Arduino like the Uno before, there are a few key differences between it and the Galileo ([Figure 1-4](#)). In fact, the specs on the Galileo make it seem like it's the product of cross-pollination between an Arduino Uno and a low-end computer.