Transducer Modules & Communication

While some of the TensorFlow Lite Micro capable development boards feature on-board sensors, others do not and you may find yourself interested in connecting an external sensor or actuator transducer module. Further, your use case may require you to periodically transmit results to other devices using wireless communication protocols like Bluetooth. In this reading, we briefly discuss each of these topics at a high level, leaving more thorough explanations to external readings you can complete at your discretion.

Sensor and Actuator Modules

When you write code to compute the average of a set of variables, you rarely if ever add the terms and divide by their number. Instead, you call on a pre-built average function. Much in the same way, many of the desired circuits required to achieve sensing and actuation of various kinds have been commoditized to the point where it isn't a question of designing these circuits again from first principles but interfacing the modules with a MCU using an API mindset, if you will. The inputs and outputs of various modules are our concern.

We'll turn to consider possible interfaces in just a moment, but first we want to highlight that there is a catalog of largely plug-and-play sensors and actuators that will work with the Arduino Tiny Machine Learning Kit, recalling that the role that each category plays in a canonical embedded system (sense → process → actuate), converting energy from and to physical phenomenon, respectively, in a process called *transduction*.

The modules we mention are available from SeeedStudio, given their compatibility with the Grove connectors on the Tiny Machine Learning Shield and are as such the easiest way to extend the functionality of your system, but there's no reason why you couldn't leverage a solderless breadboard and some hookup wire to unlock compatibility with an enormous number of available modules. In the US, we like to source such modules from Adafruit and SparkFun, but they are widely available if you do some digging.

IO & Serial Protocols

A microcontroller has three main types of IO: digital, analog, and serial, listed in order of relative complexity. The foremost is very simple, as digital pins have only two states, high (or equivalently 1 or true) and low (or 0, false), and can be written to as outputs or read from as inputs. For clarity, these binary states are physically conveyed in voltage extremes, if you will: either 0V with respect to circuit "ground" or the full-extent of the logical reference voltage for the given microcontroller, for the nRF52840, this is 3.3V.

Analog inputs are bit more complex in that they seek to translate a continuous representation of voltage in time and amplitude into a binary representation, that requires discretization and quantization, respectively. For more on analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC), including approximations made by pulse-width modulation (PWM) for slow-response loads or circumstances, you can read on at the links above.

Finally, we turn our attention to serial communication protocols, of which we'll highlight three: universal asynchronous receive transmit (UART), inter-integrated circuit (I2C), and serial peripheral interface (SPI). We've provided a brief appendix document describing them in more detail and note at a high level that the various serial protocols are designed for transmitting longer data messages with different applications in mind (e.g., one to one vs. one to many communication). We also note that variants of the I2C protocol are used by both the IMU on your Arduino Nano 33 BLE Sense and between the Arduino and the OV7675 camera module.

Efficient Wireless Communication

A major incentive for deploying deep learning models locally within an embedded system is to avoid the energetic cost of transmitting a constant stream of data from a sensor via some wireless communication interface, like WiFi or Bluetooth. Indeed, in most cases, it is favorable to compute a result that you might later transmit periodically, cutting down on the overall energy expenditure. This paradigm fits well within the dogma that led to the division of Bluetooth specification in 2011 and the creation of a new fork dedicated to transmitting information as needed, rather than in a continuous stream, Bluetooth Low Energy (BLE).

There is a BLE module, the U-Blox NINA B306, on your Nano 33 BLE Sense development board, which we will use in the Magic Wand example later in this course! If you'd like to learn more about BLE you can check out this short appendix document we created.