Computing with AI
Introduction to Robotics and IoT
(Year 1)

Course code: CC4003NI

Lecture 9

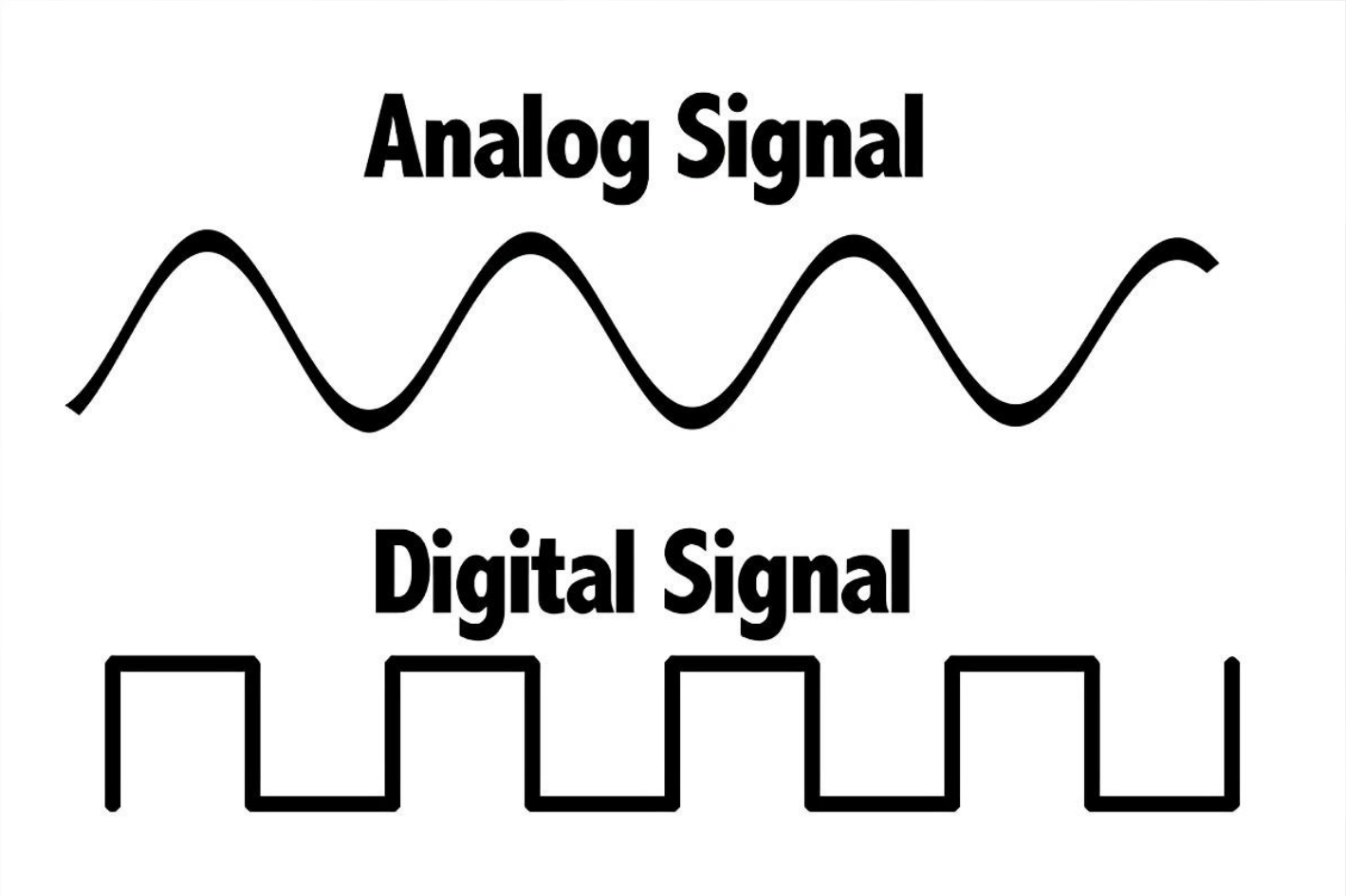
Arduino Hardware & IDE

(nabin.acharya@islingtoncollege.edu.np)

Course Outline

- Digital vs Analog signal
- Input vs Output
- Arduino Hardware
- Power pin
- Communication Interface
- Arduino Program Structure
- Writing comments
- Declaring variable and constant
- Functions
- Sample program

Digital vs Analog Signals



Inputs vs Output

- Always referenced from the perspective of the Arduino.

  Input is a signal / information going into the board.

  Examples:

  Buttons, Switches, Light Sensors, Temperature Sensors, Humidity Sensors ...

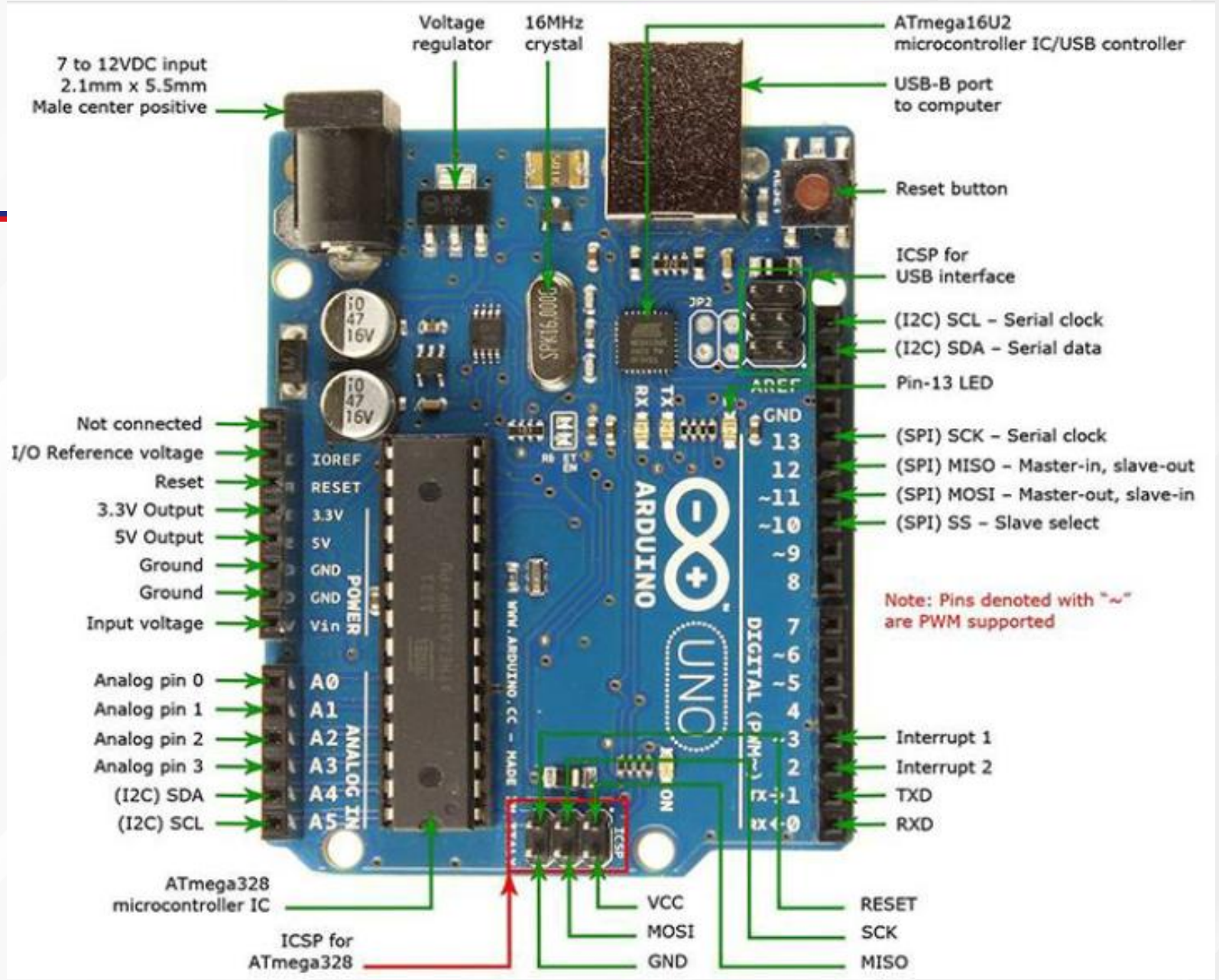
  Output is any signal exiting the board.

  Examples:

  LEDs, DC motor, servo motor, a piezo buzzer, relay, an RGB LED, ...

Arduino Hardware

- **The Arduino Uno board has:**
  - 14 digital input/output pins (of which 6 can be used as PWM outputs),
  - 6 analog inputs,
  - a 16 MHz crystal oscillator,
  - a USB connection,
  - a power jack,
  - an ICSP header,
  - and a reset button.



Summary specification

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-9V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) (0.5 KB used by bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Power pin

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

Communication Interface

- The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX).
- An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer.
- The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required.
- The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.
- The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1)

Arduino Programming

- As we learned last week, IDE stands for Integrated Development Environment.
- The IDE is a text editor-like program that allows us to write Arduino programs.
- Arduino IDE is intentionally streamlined to keep things as simple and straightforward as possible.
- Programs written for Arduino are sometimes referred to as a “sketch”.
- Part of the job of the IDE is to take the human readable code and translate it into machine-readable code to be executed by the Arduino. This process is called compiling.

Arduino Program Structure

```
/* Comments .... */

const int LED = 13;  // declaration of variables and constants

void setup()
{
....
}

void loop()
{
....
}
```

Writing Comments

- Comments are what you use to annotate your code. A well written program should include plenty of useful comments.
- Comments are meant to inform you and anyone else who might read your code. A good comment would help the reader understand how your program actually works.
- There are two ways to write comments in C:

```
// using Double Forward slash for single line comments

/* Or using a slash * combination for multi-line comments like this .... */
```

Writing Comments

- A good comment would be something like this:

  int LEDpin = 9 // This is the pin on the Arduino that the LED is plugged into

- Later on when you read your code, you will know that LED needs to be connected to pin 9, for your program to work correctly.
- Comments will be ignored by the compiler – so you can write whatever you like in them. If you have a lot you need to explain, then it may be best to use:

  /* multi-line comments */

- Comments are like the footnotes of code, except far more prevalent and not at the bottom of the page.

Declaring Variable and Constant

const int LED = 13;

int X = 200;

char Y = 5;

- A semicolon needs to follow every statement written in the Arduino programming language.
- A semicolon is part of the Arduino language syntax, the rules that govern how the code is written. It is like grammar in writing.

Functions ()

- Functions are pieces of code that are used so often that they are encapsulated in certain keywords so that you can use them more easily.
- Functions in C is very similar to mathematical functions such as:

$$y = Sin(\alpha)$$

- We give the value of angle α to the Sine function and the function returns us the value of y. We simply refer to α as the parameter or argument we pass on to the function. The value of y is then returned by the function.
- In C, arguments are passed on the function within the parentheses following the function name. Many functions require arguments to work. An argument is information the function uses when it runs.

Functions ()

- Certain functions may not need any arguments to run. In this case we may use empty parenthesis or use the reserved word: void

  For example the “delay” function as declared below:

  void delay ( );

  will not take any arguments and will not return any value.

- In Arduino, there are certain functions that are used so often they have been built into the IDE. The function pinMode(), for example, is a common function used to designate the mode of an Arduino pin.

Curly Braces {}

- Curly brackets or braces are used to enclose further instructions carried out by a function.
- There is always an opening curly bracket and a closing curly bracket.
- If you forget to close a curly bracket, the compiler will not like it and throw an error code.
- Remember – no curly brace may go unclosed!

Complete Arduino Program

```
/*
  Example 1:
  Arduino test program to an flash LED
  connected to pin 13. It simply turns the
  LED on for half a second, then off
  for half a second. And it repeats the
  entire ON/OFF cycle forever or until the
  board is switched OFF.*/

// LED is connected to pin 13:
const int LED = 13;

//the setup function runs once after a reset
void setup( )
{
 // initialize LED as an output:
  pinMode(LED, OUTPUT);
}
// the loop function runs over and over
again
void loop( )
{
  // turn the LED on (pin voltage= HIGH)
  digitalWrite(LED, HIGH);
  // wait for half a second
  delay(500);
  // turn the LED on (pin voltage= LOW)
  digitalWrite(LED, LOW);
  // wait for half a second
  delay(500);
}
```

End Of Lecture