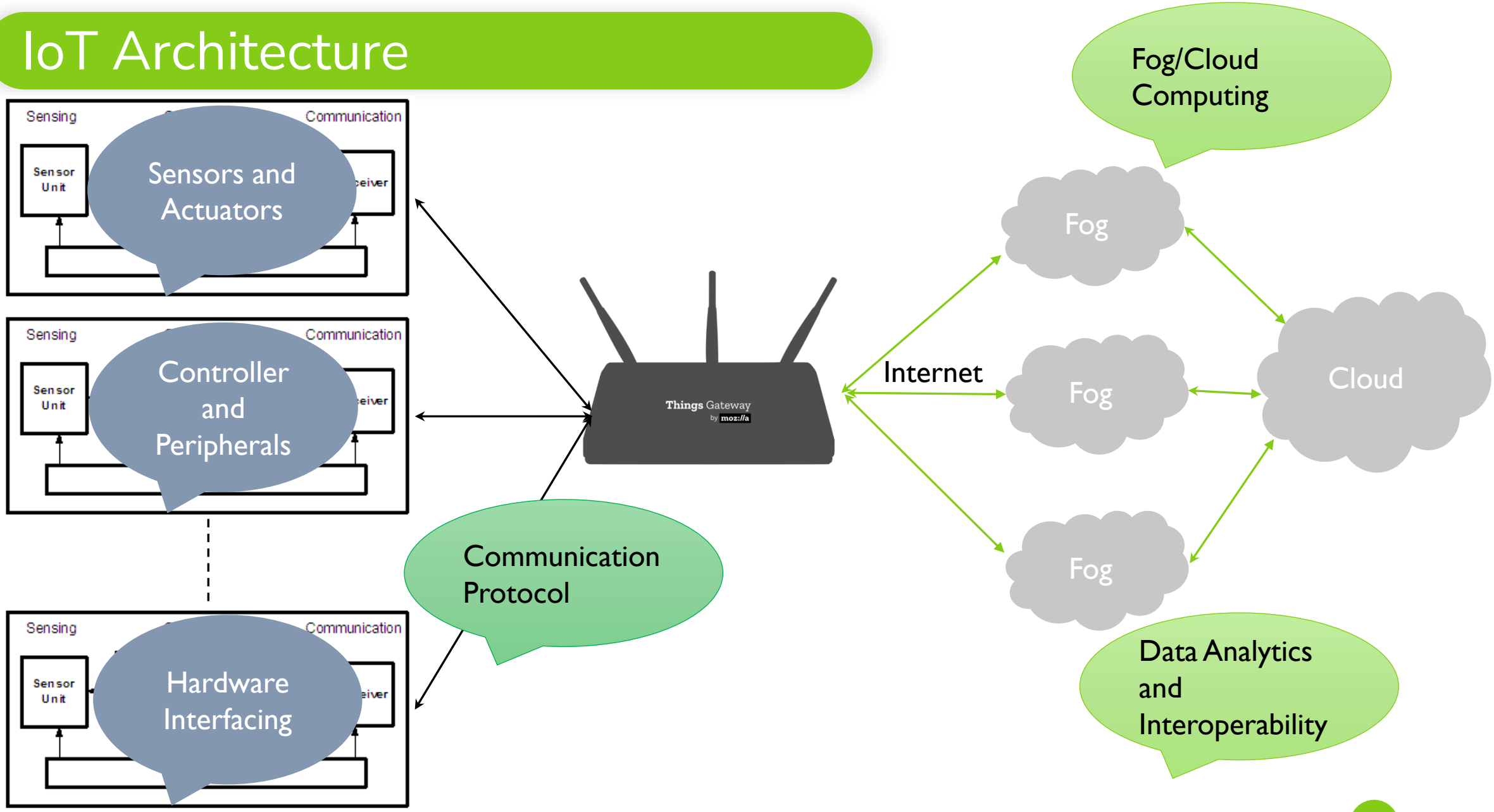
Arduino Programming

Instructor: Deepak Gangadharan

Outline

- Arduino IDE
- Arduino Programming

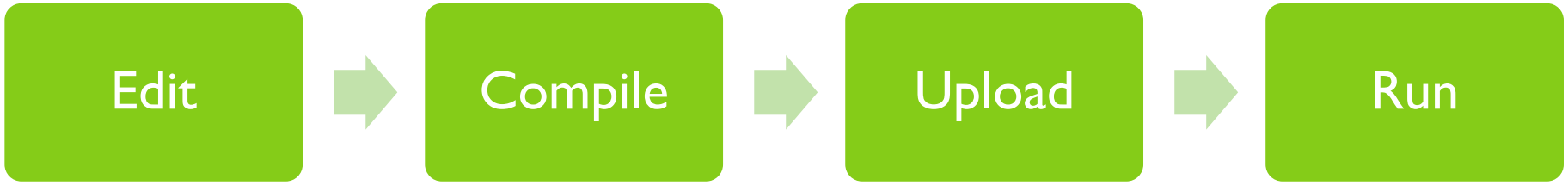
IoT Architecture



Arduino Programming

Preliminaries

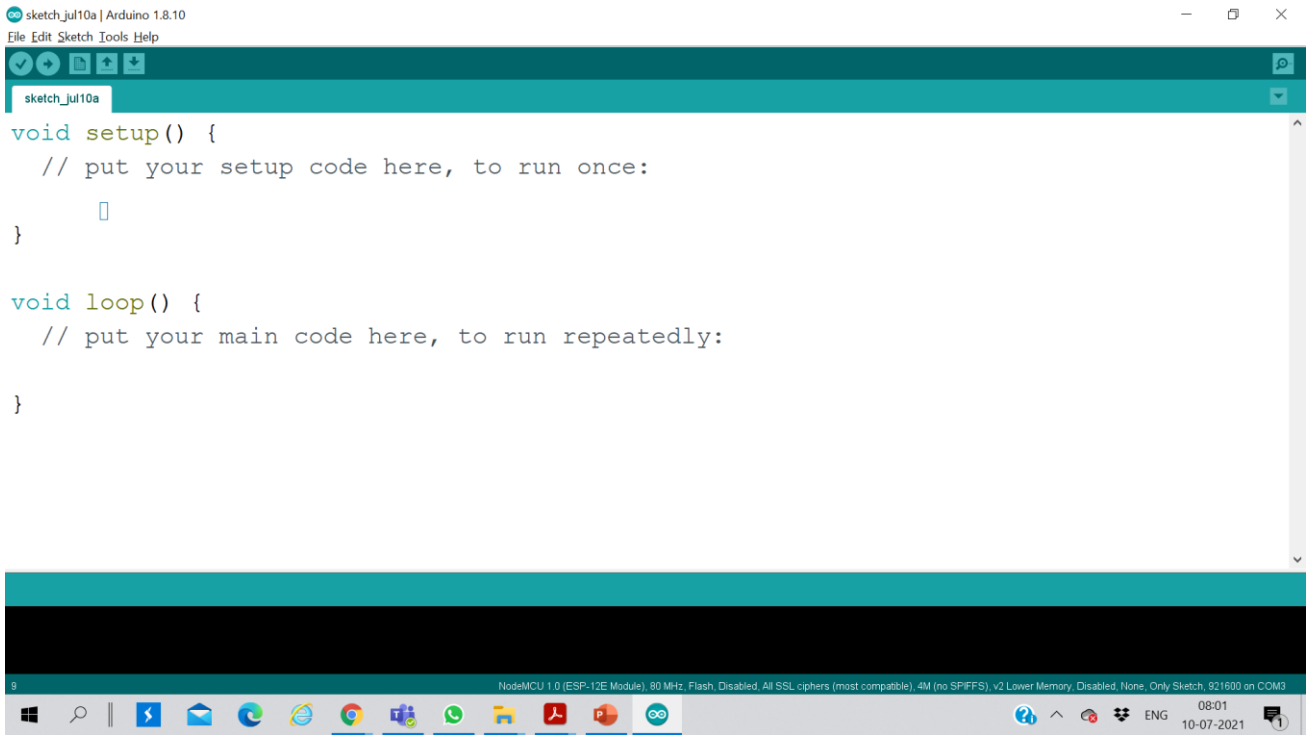
- Written in a programming language similar to C and C++
- Development cycle consists of 4 phases



- Compilation → Translates the sketch to object code
- Run → Sketch is executed as soon as upload finishes

Basic Structure

- Each sketch has 2 blocks
  - setup – Preparation block
  - loop – Execution block
- Set of statements in setup and loop enclosed within curly braces



setup and loop block

- setup
  - Initial segment of the code executed
  - Executed only once during upload/ after reset or power up
  - Initializes pin modes, libraries, variables, etc.
- loop
  - section of code executed repeatedly

Example 1

- pinMode()
  - Syntax is pinMode(pin, mode)
  - pin refers to the pin number on the board
  - mode set as INPUT or OUTPUT
- Example:
  - pinMode(10, OUTPUT) → Setting pin number 10 as output on the board
  - OUTPUT mode provides enough current to other circuits for e.g., to light a LED brightly
  - Can also use INPUT mode for e.g., in the case of a button press input

Blink

```
void setup() {
  // initialize digital pin LED_BUILTIN as an outpu
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forev
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LE
  delay(1000);                       // wait for a
  digitalWrite(LED_BUILTIN, LOW);    // turn the LE
  delay(1000);                       // wait for a
```

Example 1 (contd…)

- digitalWrite()
  - Syntax is digitalWrite(pin,value)
  - pin→ pin number or declared variable
  - Used to set a HIGH or LOW value at the pin
  - HIGH sets the value of the voltage
  - LOW sets the value to 0 or GND
- Example
  - digitalWrite(12, HIGH)
  - LED connected to pin 12 will turn ON.
- digitalRead(pin)
  - Used to read the value from a digital pin. For e.g., a sensor value

Blink

```
void setup() {
  // initialize digital pin LED_BUILTIN as an outpu
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forev
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LE
  delay(1000);                       // wait for a
  digitalWrite(LED_BUILTIN, LOW);    // turn the LE
  delay(1000);                       // wait for a
```

Example 1 (contd…)

- delay()
  - Syntax is delay(<time>)
  - A blocking function
  - <time> in milliseconds

- What is the code doing?

Blink

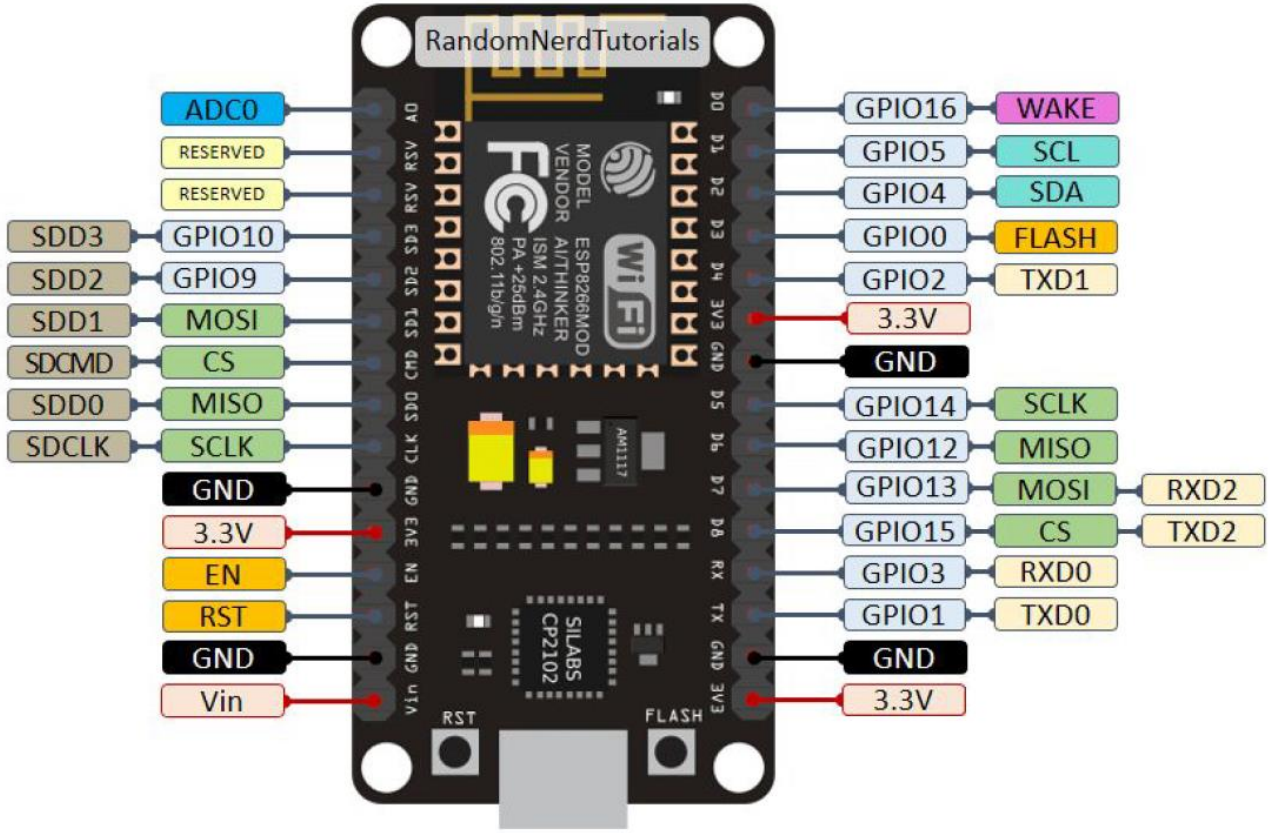
```
void setup() {
  // initialize digital pin LED_BUILTIN as an outpu
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forev
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LE
  delay(1000);                       // wait for a
  digitalWrite(LED_BUILTIN, LOW);    // turn the LE
  delay(1000);                       // wait for a
```

Syntax and Program Flow

- setup and loop do not return any value
- Compiler
  - ➢ignores spaces and tabs before code statements, in the parantheses
  - ➢ignores commas, blank lines
- Semicolon ';' used as statement termination → Compiler error if a statement found without ';'
- Program Flow: Entry point is setup() and loop() runs repeatedly

Serial Communication

- Simple communication scheme that uses UART
- Normally uses 5V for logic 1 and 0V for logic 0 → For 3.3V board uses 3V and 0V
- Messages sent to computer from GPIO1 called Tx or Transmitter
- Messages sent to Arduino from computer received on GPIO3 called Rx or Receiver
- Serial.begin() → Performs initialization to send and receive data on Rx and Tx pins
- Example: Serial.begin(14400) → 14400 is the baud rate or bps.



analogRead()

- Reads value from a specified analog pin
- ADC used to convert to discrete values
- Time required to read an analog signal on boards like UNO, Nano, etc, is 100 microseconds
- Syntax is analogRead(pin)

```
int a_pin = A1;
int value = 0;
void setup()
{
  Serial.begin(4800);
}
void loop()
{
  value = analogRead(a_pin);
  Serial.println(value);
}
```

Other Program Constructs

- Functions, Data types, Operators, Arrays are very similar as in C/C++ programming languages
- If conditional statement and loops such as for, while are also same as in C/C++
- Switch case statement is also similar as in C/C++

Arduino Interrupt

- interrupts() allow some essential tasks to run in the background
- It is enabled by default
- Disabling interrupts may impact some functions
- It may be necessary to disable interrupt in some parts of the code which are critical sections

Arduino Interrupt (contd…)

- Features
  - Monitors a user input
  - Allows processor to do something else
  - Allows quick and efficient reaction to events
- External Interrupts
  - Functions used → attachInterrupt() and detachInterrupt()

Arduino Interrupt (contd…)

- attachInterrupt()
  - Function to set the interrupt
  - Syntax is attachInterrupt(digitalPinToInterrupt(pin),ISR,mode)
  - digitalPinToInterrupt(pin) translates digital pin to interrupt number → First parameter passed to attachInterrupt()
  - To connect to pin number 4, we can use digitalPinToInterrupt(4)
  - mode → when interrupt is triggered in Arduino (LOW, FALLING, CHANGE, RISING)
  - ISR → Interrupt Service Routine, which is called when interrupts arise

Arduino Interrupt (contd…)

- detachInterrupt()
  - Function to turn off the interrupt
  - Syntax is detachInterrupt(digitalPinToInterrupt(pin))
  - digitalPinToInterrupt(pin) translates digital pin to interrupt number → First parameter passed to attachInterrupt()
  - To connect to pin number 4, we can use digitalPinToInterrupt(4)

Thank you