1. Introduction
2. Diagram
3. Types of components
4. components roles
5. pin cofigurations

- Simplified Arch
- Rpi/Arduino prgming
- Smart Arch

Here's a simple Arduino program:

cpp Copy code

```cpp
const int buttonPin = 2; // Pin connected to the button
const int ledPin = 13;   // Pin connected to the LED

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Check if the button is pressed (LOW when pressed due to pull-up re
  if (digitalRead(buttonPin) == LOW) {
    digitalWrite(ledPin, HIGH); // Turn on the LED
    delay(100); // Optional: Add a small delay to debounce the button
  } else {
    digitalWrite(ledPin, LOW); // Turn off the LED
  }
}
```

A

python Copy code

```python
import RPi.GPIO as GPIO
import time

# Set the GPIO mode to BCM
GPIO.setmode(GPIO.BCM)

# Define GPIO pins
button_pin = 17  # Change this to the GPIO pin you've connected the but
led_pin = 18     # Change this to the GPIO pin you've connected the LED

# Setup the button and LED
GPIO.setup(button_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(led_pin, GPIO.OUT)

try:
    while True:
        # Check if the button is pressed (button_pin reads LOW when pre
        if GPIO.input(button_pin) == GPIO.LOW:
            print("Button pressed! LED ON.")
            GPIO.output(led_pin, GPIO.HIGH)  # Turn on the LED
        else:
            print("Button not pressed. LED OFF.")
            GPIO.output(led_pin, GPIO.LOW)  # Turn off the LED
        time.sleep(0.1)  # Add a small delay to debounce the button

except KeyboardInterrupt:
    print("Exiting program.")
    GPIO.cleanup()  # Cleanup GPIO settings on program exit
```

1.ARDUINO

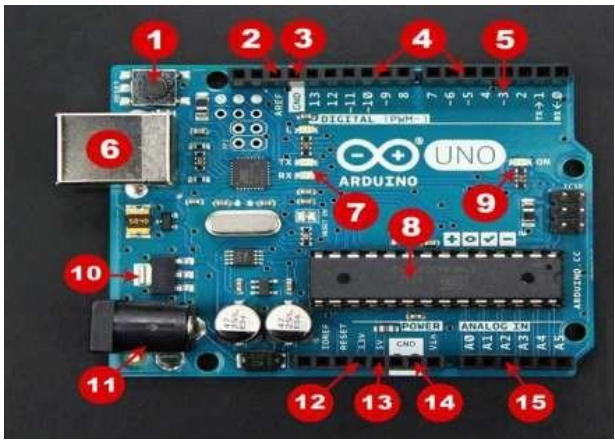
I. IOT PHYSICAL DEVICES AND ENDPOINTS - ARDUINO UNO

- Arduino is an open source advancement prototyping platform which depends on simple to utilize equipment and programming.
- Arduino is a basic single board microcontroller designed to make applications, interactive controls, or environments easily adaptive.
- Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message and turn it into an output - activating a motor, turning on an LED, publishing something online.
- The hardware consists of a board designed around an 8-bit microcontroller, or a 32-bit ARM (Advanced RISC Machines).
- Arduino Uno is a microcontroller board based on the ATmega328P.
- It has 14 digital input/output pins (of which 6 can be used as Pulse Width Modulation (PWM) outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an In Circuit Serial Programming (ICSP) header and a reset button.
- "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0.
- The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases.

II. Why Arduino?

1. Inexpensive:
   - Arduino boards are relatively inexpensive compared to other microcontroller platforms.
   - The least expensive version of the Arduino module can be assembled by hand.
2. Cross-platform:
   - The Arduino software runs on Windows, Macintosh OS and Linux operating systems.
3. Simple, clear programming environment:
   - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.
4. Open source and extensible software:
   - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries.
   - The Arduino is based on Atmel's ATMEGA microcontrollers.
   - Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.
5. Entry Level
   - Easy to use and ready to first creative projects.
   - These boards and modules are the best to start learning and tinkering with electronics andcoding.
6. Enhanced Features
   - Experience the excitement of more complex projects, with advanced functionalities, orfaster performances.
7. Internet of Things
   - Make connected devices easily with IoT and the world wide web.
8. Wearable
   - Add smartness to projects and sewing the power of electronics directly to textiles.

5.2.EXPLORING ARDUINO UNO LEARNING BOARD



1. Reset Button
   - This will restart any code that is loaded to the Arduino board
2. AREF
   - Stands for “Analog Reference” and is used to set an external reference voltage
3. Ground Pin
   - There are a few ground pins on the Arduino and they all work the same
4. Digital Input / Output
   - Pins 0-13 can be used for digital input or output
5. PWM
   - The pins marked with the (~) symbol can simulate analog output
6. USB Connection
   - Used for powering up your Arduino and uploading sketches
7. TX/RX
   - Transmit and receive data indication LEDs
8. ATmega Microcontroller
   - This is the brains and is where the programs are stored
9. Power LED Indicator
   - This LED lights up anytime the board is plugged in a power source
10. Voltage Regulator
    - This controls the amount of voltage going into the Arduino board
11. DC Power Barrel Jack
    - This is used for powering your Arduino with a power supply
12. 3.3V Pin
    - This pin supplies 3.3 volts of power to your projects
13. 5V Pin
    - This pin supplies 5 volts of power to your projects
14. Ground Pins
    - There are a few ground pins on the Arduino and they all work the same
15. Analog Pins
    - These pins can read the signal from an analog sensor and convert it to digital

III. Things that Arduino can do

1. can control with Arduino is an LED.
2. Display a message in LCD Display
3. Control DC or Servo Motors
4. Read Data from outside world
5. Motion sensor allows us to detect movement
6. Light sensor allows to measure the quantity of light outside world
7. Humidity and Temperature sensor used to measure humidity and temperature.
8. Ultrasonic sensor allows to determine the distance to an object through sound
9. Shields are an extension of the Arduino

IV. Arduino Uno programming

Arduino Uno programming is based on the Arduino programming language, which is a simplified version of C/C++. Here are some important points about Arduino Uno programming:

1. IDE (Integrated Development Environment):

Arduino Uno is programmed using the Arduino IDE, a user-friendly software that simplifies the process of writing, compiling, and uploading code to the Arduino board.

2. Setup and Loop Functions:

- ***setup() function: Runs once when the Arduino is powered on or reset. It is used to initialize variables, pin modes, and other settings.***
- ***loop() function: Runs continuously after the setup() function. It contains the main code that the Arduino executes repeatedly.***

3. Pin Modes:

- ***Use pinMode(pin, mode) to set the mode of a pin (INPUT, OUTPUT, INPUT_PULLUP).***
- ***Digital pins can be used for both input and output, while analog pins can also be used for analog input.***

4. Digital and Analog I/O:

- ***Digital I/O: Use digitalWrite(pin, HIGH/LOW) to set a digital output pin to high or low. Use digitalRead(pin) to read the state of a digital input pin.***
- ***Analog I/O: Use analogWrite(pin, value) to provide a PWM (Pulse Width Modulation) output. Use analogRead(pin) to read an analog input.***

5. Functions:

- ***Define custom functions to break down the code into modular and reusable sections.***
- ***Functions are declared with the syntax returnType functionName(parameters).***

6. Control Structures:

- ***Use standard control structures like if, else, while, for to control the flow of the program.***

7. Serial Communication:

- ***Use Serial.begin(baudRate) to initialize serial communication. Common baud rates are 9600, 115200, etc.***
- ***Use Serial.print() and Serial.println() to send data to the serial monitor for debugging.***

8. Libraries:

- ***Arduino provides a variety of libraries that extend the functionality of the board. You can include these libraries in your code using #include <LibraryName.h>.***

9. Variables:

- ***Declare variables using the syntax dataType variableName. Common data types include int, float, char, etc.***

10. Arrays:

- ***Use arrays to store multiple values of the same data type.***

11. Comments:

- ***Use comments (// for single-line comments, /* */ for multi-line comments) to add explanatory notes to your code.***

12. Power Supply:

- ***The Arduino Uno can be powered via USB or an external power supply connected to the DC power jack.***

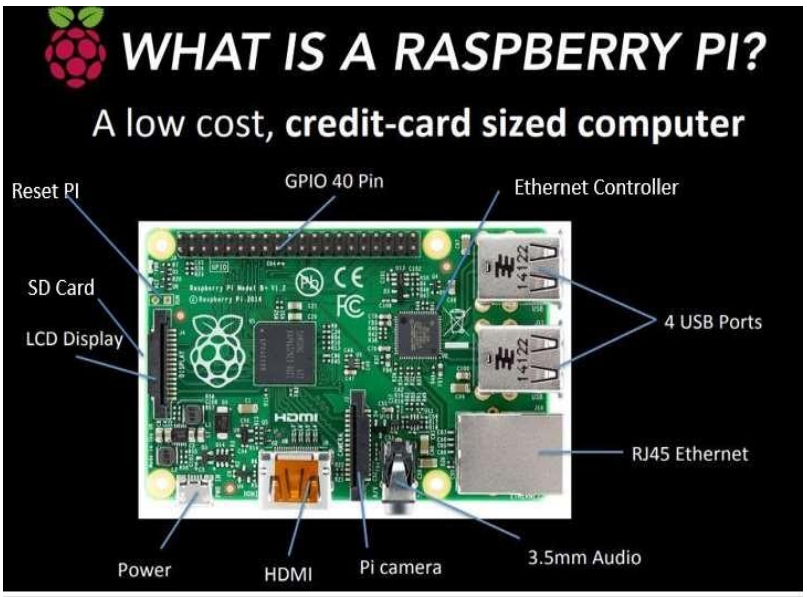
13. Common Functions:

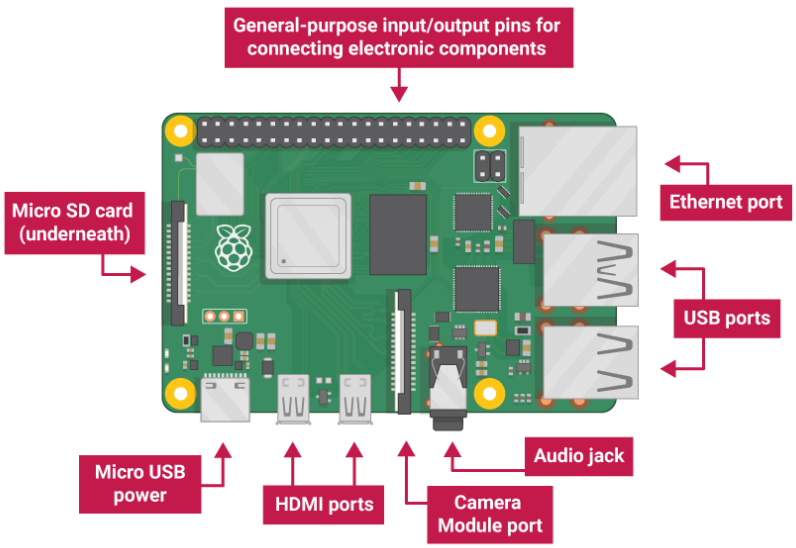
- ***delay(milliseconds): Pauses the execution of the program for the specified time.***
- ***millis(): Returns the number of milliseconds since the Arduino started running.***

Remember that Arduino programming is often used for prototyping and simple embedded systems. As you gain more experience, you can explore advanced topics such as interrupts, advanced sensor integration, and communication protocols.

2.Raspberrypi

- Raspberry Pi is a series of credit card sized board computers developed in UK by RaspberypiFoundation to promote the teaching of basic computer science in schools and developing countries.
- Generations:
- First generation: RaspberryPi 1 Model B) was released in February 2012 followed by simpler and inexpensive model A.
- In 2014, the foundation released a board with improved design in Paspberry 1 Model B+.
- Raspberry Pi is the name of a series of single-board computers made by the Raspberrypi Foundation, a UK charity that aims to educate people in computing and create easier access to computing education.
- The Raspberry Pi launched in 2012, and there have been several iterations and variations releasedsince then.
- The original Pi had a single-core 700MHz CPU and just 256MB RAM, and the latest model has aquad-core 1.4GHz CPU with 1GB RAM.
- The main price point for Raspberry Pi has always been $35 and all models have been $35 or less,including the Pi Zero, which costs just $5.





V. COMPONENTS

1. USB ports — these are used to connect a mouse and keyboard. You can also connect other components, such as a USB drive.

2. SD card slot — you can slot the SD card in here. This is where the operating system software and your files are stored.

3. Ethernet port — this is used to connect Raspberry Pi to a network with a cable. Raspberry Pi can also connect to a network via wireless LAN.

4. Audio jack — you can connect headphones or speakers here.

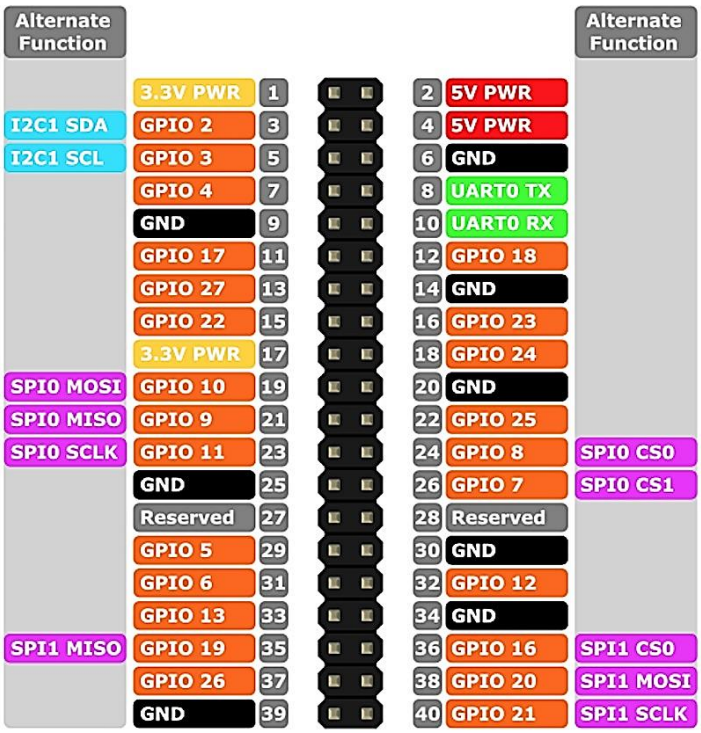
5. HDMI port — this is where you connect the monitor (or projector) that you are using to display the output from the Raspberry Pi. If your monitor has speakers, you can also use them to hear sound.

6. Micro USB power connector — this is where you connect a power supply. You should always do this last, after you have connected all your other components.

7. GPIO ports — these allow you to connect electronic components such as LEDs and buttons to Raspberry P

VI. PIN CONFIGURATION

- The pin numbers mentioned above are the Broadcom GPIO numbers.
- Some pins have alternate functions, such as I2C, SPI, UART, etc. You can configure these pins for different functionalities in your code.
- Pins 1 and 2 provide +3.3V power, pins 4 and 6 provide +5V power, and pins 9 and 14 are ground (GND).
- The GPIO pins can be controlled using programming languages like Python, and libraries like RPi.GPIO can be used to simplify the process.



3.Introduction to Smart City IoT Architecture:

Smart City IoT (Internet of Things) architecture is a framework designed to integrate technology and connectivity into urban environments to enhance efficiency, sustainability, and the overall quality of life for citizens. In a Smart City IoT system, various devices and sensors are deployed throughout the city to collect data, monitor activities, and enable intelligent decision-making. This data is then processed and utilized to provide innovative services, optimize resource usage, and improve urban planning.

I. Street Layer:

1. ***Composed of devices and sensors collecting data and executing actions based on instructions.***
2. ***Sensors include magnetic sensors for parking events, lighting controllers, video cameras for traffic and security, air quality sensors, and device counters.***
3. ***Sensors play a crucial role in collecting data for various smart city use cases.***

II. City Layer:

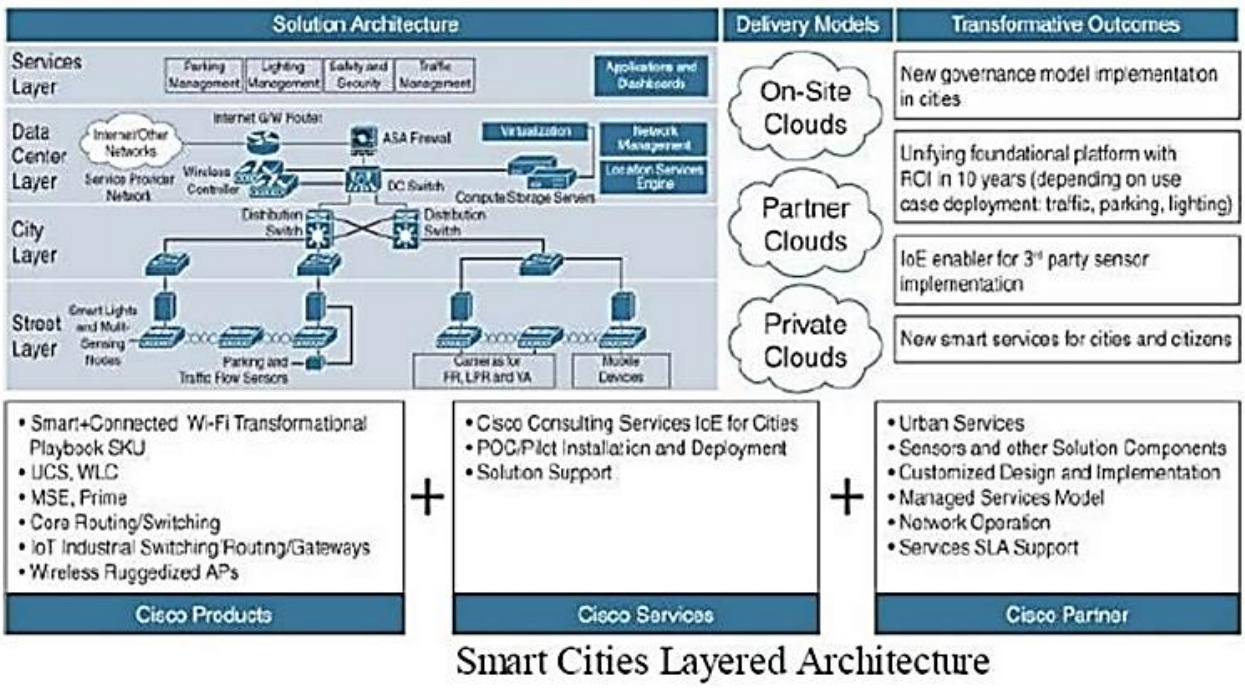
1. ***Located above the street layer, it deploys network routers and switches to handle the city's data transportation needs.***
2. ***Aggregates data collected by sensors into a single transport network.***
3. ***Must support multiple protocols for different IoT applications.***
4. ***Emphasizes resiliency to ensure successful data forwarding, with redundancy paths using protocols like Resilient Ethernet Protocol (REP).***

III. Data Center Layer:

1. ***Receives data from sensors for processing and correlation.***
2. ***Processes data to derive meaningful information and trends.***
3. ***Utilizes cloud infrastructure for storage, virtualization, adaptability, and analytics capabilities.***
4. ***Enables real-time data processing, which traditional city networks may struggle to handle.***

IV. Service Layer:

1. ***Focuses on delivering value to different users within the city, including operators, citizens, and law enforcement.***
2. ***Customizes data visualization based on specific user needs and use cases.***
3. ***Addresses data security concerns by implementing security protocols at each layer.***
4. ***Security features include device/sensor identification and authorization, encryption, and user authentication.***



Smart Cities Layered Architecture