**Characteristics of Pervasive Computing**

• Embedded: networked devices integrated into the environment

• Context aware: these devices can recognize you and your situational context

• Personalized: they can be tailored to your needs

• Adaptive:they can change in response to you

• Anticipatory(预测):they can anticipate your desires without conscious mediation.

• Natural User Interface (NUI): voice, gestures

**Pervasive Computing Technology**

•Hardware: microcontrollers, small systems, cloud systems

•Sensors and actuators:Digital/analogue, wearable

•Interfaces: parallel, serial, media

•Software:programming languages,IDEs, cloud based development environments, platforms

•Networks: hardware/software/protocols, WiFi,

Bluetooth, Ethernet, mobile networks, NFC...

• Packaging: materials, 3D printing, design

**Internet of Things(Definition&Components)**

The Internet of things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.

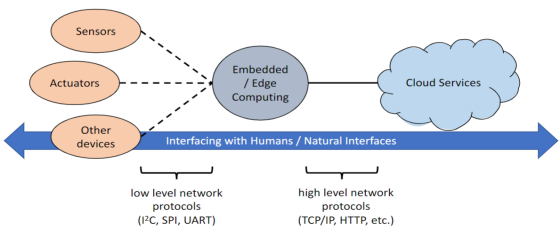
• Smart Objects (Devices and sensors):

Embedded computing elements capable of collecting and storing data

• Connected: Able to transmitting and receiving information through different connectivity mediums (e.g. Bluetooth, WiFi, RFID, etc.)

• Information System Infrastructure: A system for managing and making use of the objects and data

• User Interface: User interfaces required to present the information to user



**Application areas of IoT**

Consumer:Smarthomes, healthcare,transport

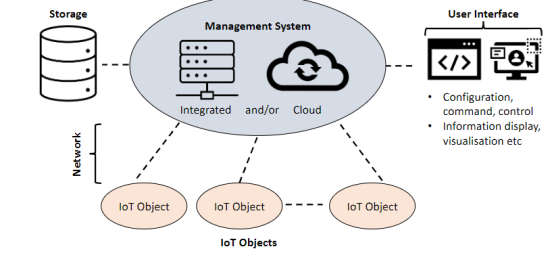
Commercial: Building management, transport(monitoring and control), retail.

Industrial: Factory monitoring and control.

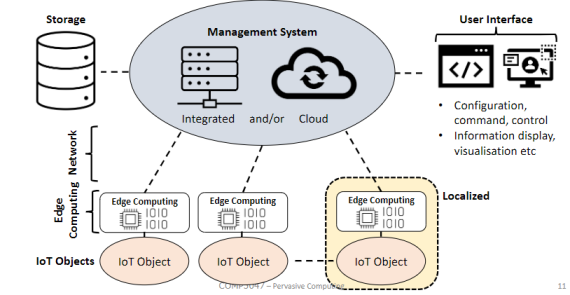
Agricultural: Crop monitoring

Government: Smart cities.

**Basic Structure for IoT**



**Edge Computing Distributed**



**Weiser Paper**

Seminal paper

“Ubicomp” (Ubiquitous computing) ...

Hardware and networking

Software for personalisation and context

Broader societal and multi-disciplinary research, practice, commerce

**Managing IoT system:**

* IoT systems are widespread and located close to users, making them challenging to control.
* There's a need for innovative management methods.
* Management can be central (e.g., software updates or power-saving settings for multiple devices) or controlled by end-users.
* End users might adjust settings based on preferences, like setting fitness goals or optimizing home lighting. They interact with these systems through user-friendly interfaces, which can include touch or voice commands.

**End User "Programming"**

* Traditional coding methods aren't suitable for most users.
* Most people can't code in traditional languages.
* Solutions involve visual or graphical programming tools.
* Users set up simple logic like: "If X happens, do Y."
* This can chain multiple rules together.
* These systems rely on sensor input and can control displays and other devices.

Examples:Node-red:Provides graphical program to design data flow.

**Analog vs Digital**

Digital Signals:

数字信号通常由一系列的0和1表示，如电压

它们在性质上是离散的，主要可以表示为二进制形式。

Analog Signals:

模拟信号是连续的，其幅度和时间都是变化的。

它们受到许多波动和变化的影响。

Analog Vs. Digital:

模拟信号容易受到干扰，有更高的观测误差，并且消耗更多的电力。但是，它们可以使用较小的带宽实时处理。

数字信号受噪声的影响较小，观测误差低，更能有效地节省电力，但可能需要更多的带宽，且不总是能保证实时处理。

**UART/COM Port**

* Universal Asynchronous Receiver-Transmitter; a serial communication protocol.
* Before USB: Primary means of device-to-computer communication.
* Type: Digital interface; full-duplex
* Usage: Common in embedded systems, small devices, industrial communications, and instruments.
* Range: Limited by cable length.
* Communication:Point-to-point; generally lacks multipoint capability.
* Synchronization:Asynchronous需要波特率同步信号，不按照钟表时间同步
* Baud Rate:Determines transfer speed; higher rates more noise-sensitive.

**Analog to Digital Converter (ADC)**

ADCs are designed for analog signals with limited bandwidth.（模拟信号转换为数字信号

* The ADC process involves: 1) Sampling, 2) Quantizing, 3) Encoding.
* According to the Nyquist–Shannon sampling theorem: If a function x(t) has frequencies no higher than B hertz (Hz), an appropriate sampling rate is anything larger than 2B samples per second.
* Sampling below the Nyquist rate leads to aliasing(混叠).
* Sampling too quickly can result in: excessive(过度) data, high power consumption, and the need for costly hardware.

**Digital IO (Input/Output):**

Refers to microcontroller interactions with external components.

Common in devices like Arduino.

Input:

* Capability to read an external pin's status.
* Uses threshold voltages to determine HIGH or LOW.
* Can only detect HIGH or LOW, no in-between values.

Output:

* Ability to send voltage to external components.
* HIGH voltage level varies (e.g., 5V or 3.3V) depending on device.
* Only two output states: HIGH or LOW.

**Pulse Width Modulation (PWM):**

Basics:

* Technique for controlling output intensity in digital devices.
* Provides control beyond traditional binary (0/1) signals.
* Uses varying pulse widths in periodic digital signals.
* Underlying Principle: Relies on "Persistence of Vision" — eyes average brightness over time.

Applications:

* Intensity control (e.g., LED dimming).
* Communication protocols.
* Robotics (e.g., servo motor control).

Arduino Usage:

* Simple implementation with analogWrite(pin, value).
* PWM on ESP microcontrollers is more complex.

**Serial Peripheral Interface (SPI):**

* Synchronous Communication: Uses a clock signal to synchronize data transfer between master and slave devices.
* Reduced Overhead: Achieves higher data rates by overcoming the overheads of asynchronous communication like UART.
* Master-Slave:Configuration: Communication involves one master device and one or multiple slave devices.
* Full Duplex: Supports simultaneous data transmission and reception.

Examples of SPI Communication:

* Microcontroller & Memory: SPI is often used for communication between microcontrollers and external Flash memory or EEPROM.
* Sensor Reading: Some temperature or pressure sensors communicate their measurements to the main controller via SPI.
* LCD Control: Certain LCD displays receive commands and data from microcontrollers through SPI.
* Wireless Modules: Some Wi-Fi or RF modules use SPI for data exchange with the main controller.

**I2C (Inter-Integrated Circuit):**

* Synchronous Communication: I2C is a synchronous protocol, using a clock signal to synchronize communication
* Device Addressing: Each I2C device has a unique address, eliminating the need for multiple select lines as in SPI.
* Half-Duplex
* A strength of I2C is that adding more devices doesn't drastically(猛烈的) increase complexity. More devices can lead to reduced data speeds..
* Noise Concerns: Adding more devices can introduce electronic noise, potentially leading to communication errors.

**Open Systems Interconnection (OSI):**

* OSI is like a rulebook that explains how different computer systems can talk to each other.
* It doesn't care about the tech details inside each system; it focuses on communication between systems.
* It helps different computer systems work together smoothly by wrapping and unwrapping data in a specific way.

1. Physical Layer

* Deals with the medium, signal, and binary transmission.

1. Data Link Layer

* Responsible for physical addressing.

1. Network Layer

* Handles path determination and logical addressing.

1. Transport Layer

* Provides end-to-end connections and reliability.

1. Session Layer

* Interhost communication

1. Presentation Layer

* Manages data representation and encryption.

1. Application Layer

* Connects network processes to end-user applications

数据发送时：从上到下，每层添加该层相关信息

数据接收时：从下到上，每层移除或解释对应的头部尾部信息

每层只需要直到上一层的接口和服务，例如，传输层不需要知道数据将通过什么类型的物理网络发送（如铜线、光纤或无线），它只需要关心如何与网络层交互，以确保数据正确无误地传输到目的地

**Network Topologies**

Point-to-Point: 

The simplest form of connection.

Connects only two devices.

Wiring is straightforward.

Line Topology: 

An extension of point-to-point.

If one node fails, it could disconnect the entire network.

Ring Topology: 

Direct device-to-device connection.

With bidirectional links, a single device failure won't disrupt the entire network.

Bus Topology: 

Allows connection of multiple devices.

Performance is determined by the bus's speed and reliability.

Star or Hub-and-Spoke Topology: 

Connects multiple devices to a central hub.

Scalability is limited due to the need for multiple lines to the hub.

Mesh Topology: 

Can be either fully or partially connected.

Offers redundancy.

Wiring is complex, and debugging can be challenging.

Tree Topology: 

Uses hierarchical wiring.

Suitable for edge computing.

Failure at a hub could lead to loss of an entire branch.

Hybrid Topologies: 

Most real-world networks are a mix of different topologies.

**Technologies for Networking**

Wired Networking:

• COM port,ADC / Digital IO / PWM,SPI, I 2C,Ethernet,USB,Optical

• Communication over power lines

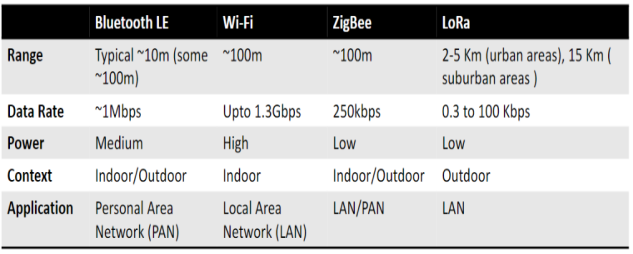
Wireless Networking:

• Bluetooth,Wi-Fi,Zigbee,LoRa,Analog

• E.g. VHF/UHF TV, AM/FM Radio

• Digital

• 3G/ 4G / 5G / LTE-M / Z-Wave



**Higher Level Communication Protocols**

• IoT devices are powerful enough to use high level protocols

• Web – HTTP

Hypertext Transfer Protocol (HTTP)

* Application layer protocol
* Text based – lines of text
* A message has two parts
* header and body separated by a blank line
* The body has content.
* Mainly for web content, but now used for others

**Evaluation**

Technical Evaluation:

Technical specification

• Power consumption

• Size / Weight / Materials

Lifecycle:

Assess the longevity and reliability of the wearable sensor, often requiring automated setups.

Safety:

Evaluate electrical safety, material safety, and potential radiation or electromagnetic wave emissions,data safety

User Evaluation:

Usability of the System:

* Assess how easy and helpful the wearable sensor is to learn and use.
* Focus on the user-friendliness of the interface and interactions.

User Experience:

* Examine the users' satisfaction, enjoyment, and motivation when using the device.
* Determine if the sensor provides a positive and engaging experience for the target user population.

Evaluating with Users:

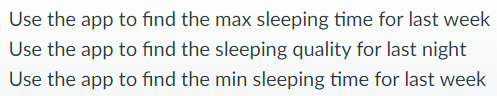
* Recognize that the designer's perspective may differ from users' opinions.
* Evaluation ensures that the design is appropriate and acceptable for the intended users.

**Abstract Tasks & Concrete Tasks**

Abstract tasks

E.g., The user can navigate the app

Concrete tasks

E.g., Find the total number of steps the system recorded in the past 7 days

**Interaction type**

指导 (Instructing):用户告诉系统该做什么。例如: 键入搜索关键词、点击保存按钮

对话 (Conversing):用户与系统双向交流。例如: ATM、提问并获得答案。

操纵 (Manipulating):用户直接操作界面元素。例如: 用手指缩放图片、拖动图标。

探索 (Exploring):用户探索系统信息或功能。例如: 在地图中浏览、VR

响应 (Responding):系统反馈用户的行为或请求。显示提交确认、点击播放音乐。

**Interface Type:**

命令行界面 (CLI):通过文本指令与计算机交互的界面。Windows中的命令提示符

图形用户界面 (GUI):通过视觉元素（如图标和窗口）与计算机交互的界面。

例：Windows操作系统的桌面。

后WIMP用户界面:超越传统“窗口、图标、菜单、指针”(WIMP)的交互界面。

例：多点触控屏幕的捏合放大手势。

语音界面:通过语音命令与计算机或设备交互的界面。Apple的Siri。

交互式表面:能够检测和响应用户触摸或近距离交互的物理表面。Microsoft Surface

有形界面 (TUI):通过物理对象表示和控制数字信息的界面。Reactable数字音乐

虚拟现实 (VR):计算机生成的仿真环境，其中用户可以与其互动。Oculus Rift。

可穿戴界面:可以穿戴并在移动时使用的设备界面。Apple Watch。

**ADC**

#include <Arduino.h>

int lightSensor = 0; //make a variable for

void setup()

{

pinMode(A0, INPUT); // Setup pin to be an input

}

void loop()

{

lightSensor = analogRead(A0); // read the ADC value

delay(200); // you can control the speed of sampling here

}

**PWM**

#include <Arduino.h>

// the number of the PWM pin

const int pwmPin = D9; // D9 the LED pin

// setting PWM properties

const int freq = 5000; // how fast the PWM signal work (cycle = 1 / freq)

const int pwmChannel = 0; // which internal pwm channel to use

const int resolution = 8; // what resolution (8 bits means -> 2^8, 255 levels)

int brightness = 0; // variable to hold brightness (0 off, 255 full power)

void setup()

{

pinMode(pwmPin, OUTPUT);

// configure PWM functionalitites

ledcSetup(pwmChannel, freq, resolution);

// attach the channel to the GPIO to be controlled

ledcAttachPin(pwmPin, pwmChannel);

ledcWrite(pwmChannel, brightness);

}

void loop()

{

// lets gradually increase and decrease the brightness

for(int i=0; i < 100; i++){

brightness = i \* 255 / 100; // increase brightness with i

ledcWrite(pwmChannel, brightness); // set brightness as pwm

delay(10); //

}

for(int i=0; i < 100; i++){

brightness = (100 - i) \* 255 / 100;

ledcWrite(pwmChannel, brightness);

delay(10);

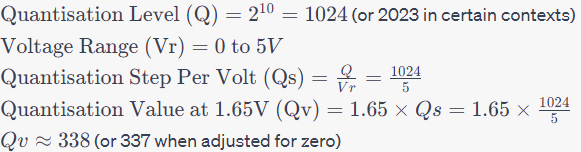
}

}

**Quiz**

The CS pin in the Serial Peripheral Interface (SPI) is used to select the peripheral interface the controller is communicating with

The Analog to Digital Converter (ADC) functions within an input range of 0 to 5V and has a bit resolution of 10 bits. If we provide the ADC with an input voltage of 1.65V, we want to determine the approximate integer output it would generate, such as what we would get from a function like analogRead().



With respect to the basic structure for IoT,

[]Edge computing creates a distributed system. / Storage can be physically connected to IoT objects.

With respect to the management system of IoT

[]A change that applies to many IoT objects can be centrally controlled. / IFTTT is a good method for end-user programming because it is easy to code with.

When you use an analog sensor like a force sensitive resistor (FSR) with ADC, a series resistor must be carefully chosen so that:You convert the resistance change to a voltage change

You are asked to characterize a digital wearable sensor that measures the elbow angle of the wearer. Your goal is to validate that it has an angular resolution of +/- 4 degrees. Select the possible state of the art choice(s) to validate your sensor.

[]An analog elbow angle sensor with +/- 1 degrees/A digital elbow angle sensor with +/- 1 degrees