Introduction To Embedded System

A BASIC OVERVIEW OF MICROCONTROLLER, SENSOR, DISPLAY, COMMUNICATION PROTOCOL AND M5STACK CORE 2

Overview

- What is an Embedded System?
- Basic components of an Embedded System
- Simplified overview of microcontroller, sensor and display
- GPIO and communication protocols (UART, I2C and SPI)
- Introduction to M5Stack Core 2

What is an Embedded System?

- A specialized computing system designed to perform a specific task
- Combines hardware (e.g., microcontroller, sensors) and software (e.g., firmware)
- Common in everyday devices: smart watch, smart TV, microwave oven, etc.
- Car control systems, medical devices, robotics, etc. are some examples

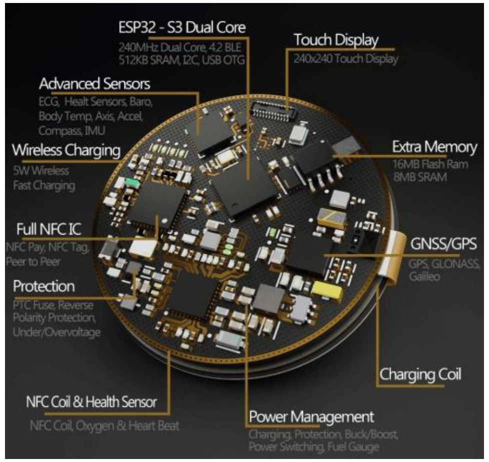

Fig: Smart Watch – An embedded System

Microcontroller: The Brain of Embedded Systems

- A microcontroller (MCU) is a small computer on a single chip
- It has a CPU, memory, and programmable input/output peripherals
- Common microcontrollers: ATmega328p, STM32, ESP32, PIC
- Typically used in embedded systems for control and data processing tasks

Fig: A STM32F103 microcontroller

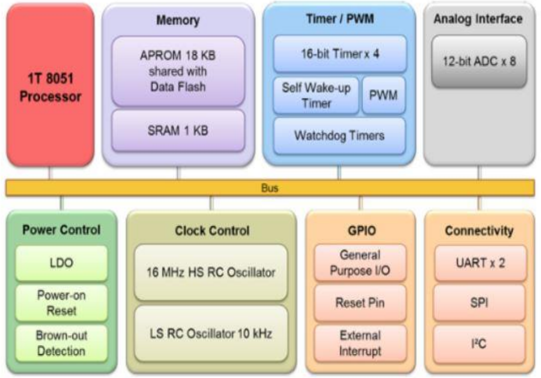

Fig: Block diagram of an 8051 microcontroller

Sensors in Embedded Systems

- Sensors detect changes in physical or environmental conditions
- Sensors convert real-world data into electrical signals
- Types of sensors: Temperature, humidity, pressure, motion, light, etc.
- Example: A temperature sensor (e.g., DHT11) converts temperature and humidity into digital data for a microcontroller

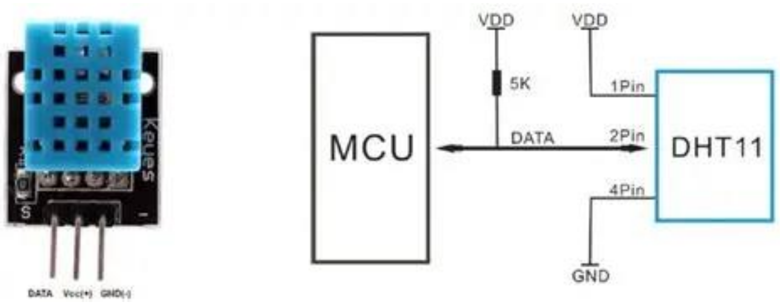

Fig: A DHT11 sensor is connected with a microcontroller

Display Devices in Embedded Systems

- Displays are used to show information or feedback in embedded systems
- Types of displays: LCD, OLED, 7-segment, TFT, etc.
- Often used in devices like calculators, clocks, or digital thermometers
- Displays communicate with microcontrollers via protocols like SPI or I2C

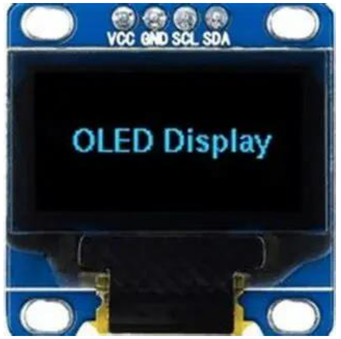

Fig: An OLED display

General Purpose Input/Output (GPIO)

- GPIO pins allow microcontrollers to interface with the external world
- Can be configured as input (e.g., reading sensors) or output (e.g., controlling LEDs)
- GPIO pins are the fundamental way to interact with devices in embedded systems
- Example: Turning an LED on/off using GPIO pins

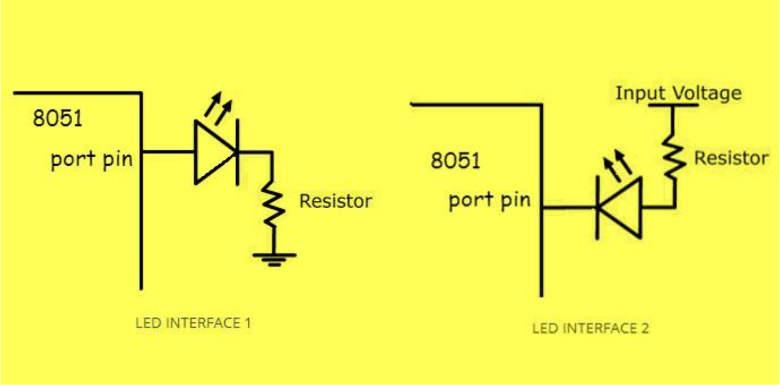

Fig: A LED is connected to a microcontroller GPIO

Universal Asynchronous Receiver-Transmitter (UART) Protocol

- Simple, asynchronous and full-duplex
- Generally, uses TX (Transmit) and RX (Receive) and no clock is required
- GPS, Bluetooth, Wi-Fi modules, etc. use UART to transmit data
- Commonly used to communicate with a PC. Also used for debugging

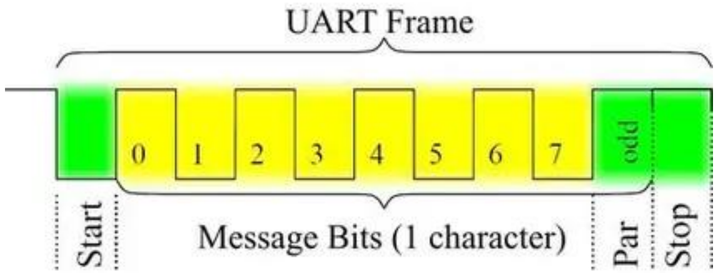

Fig: A UART frame

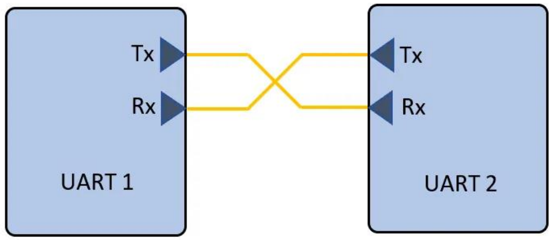

Fig: UART communication between two devices

Inter-Integrated Circuit (I2C) Protocol

- A synchronous communication protocol
- Uses only two wires: SDA (Serial Data) and SCL (Serial Clock).
- Multiple devices with unique addresses can be connected up to a limit.
- Used in interfacing sensors, real-time clocks, displays, EEPROM etc.

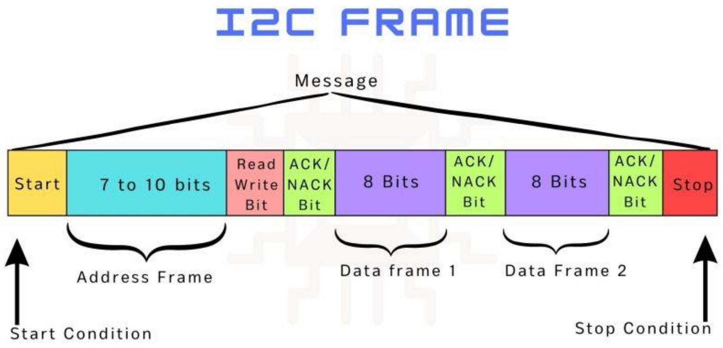

Fig: An I2C frame

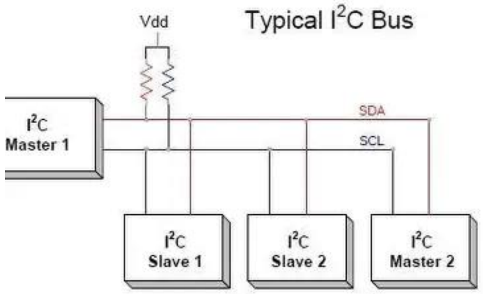

Fig: A I2C bus with multiple devices

Serial Peripheral Interface (SPI) Protocol

- A synchronous, full-duplex high-speed communication protocol
- Uses four wires: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock), and CS (Chip Select).
- Data flows simultaneously in both directions (MOSI & MISO).
- Common Uses: Flash memory, SD cards, sensors, displays, and other high-speed devices.

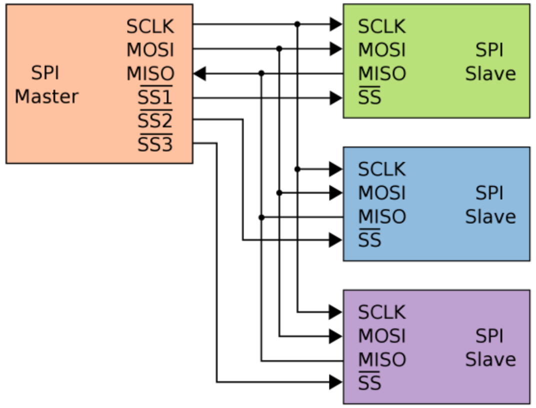

Fig: Devices are connected through SPI

M5Stack: A Powerful Embedded Development Platform

- M5Stack is a modular and stackable embedded development kit.
- Based on the ESP32 microcontroller, it provides Wi-Fi and Bluetooth connectivity.
- Comes with various built-in sensors, displays, and modules.
- Ideal for rapid prototyping and IoT projects.
- Supports programming in Arduino IDE, MicroPython, and more.

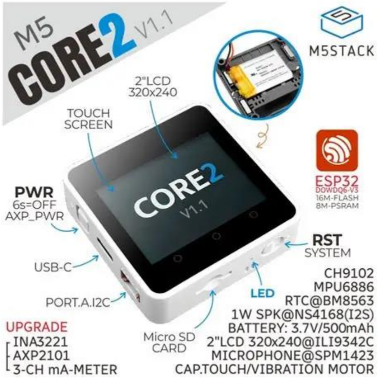

Fig: M5Stack Core 2

Thank You

Any Questions?