EMBEDDED SYSTEMS DESIGN

MODULE 2

Memory, Sensors, and Actuators: ROM, RAM, Memory according to the type of Interface, Memory Shadowing, Memory selection for Embedded Systems, Sensors, and Actuators (I/O components): relay, Motors with PWM control, stepper motor. Other Sub-sub-systems: Timer, counter, Reset Circuit, Brown-out Protection Circuit, Oscillator Unit, Real Time Clock, and Watchdog Timer

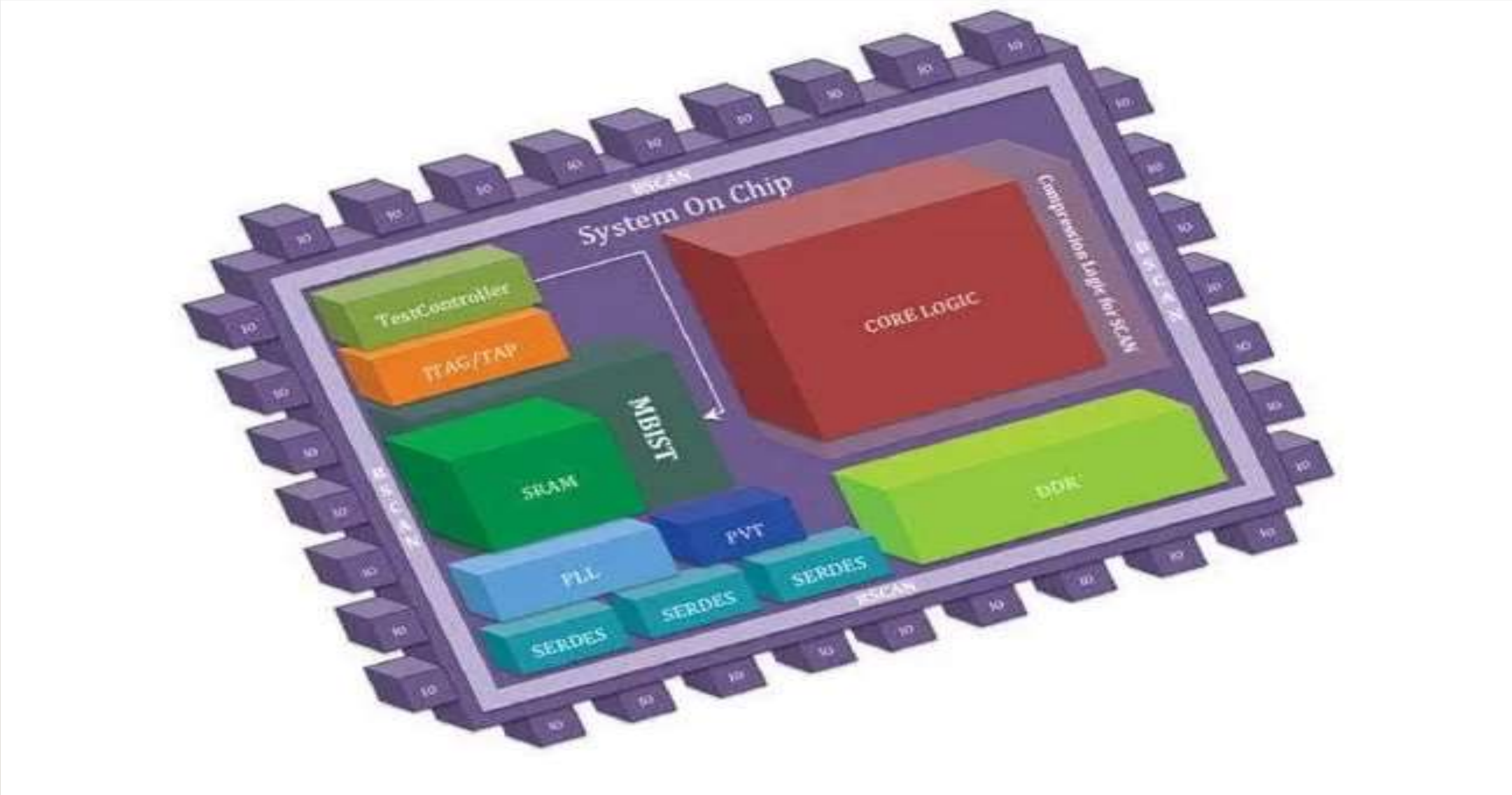
INTRODUCTION TO EMBEDDED SYSTEMS

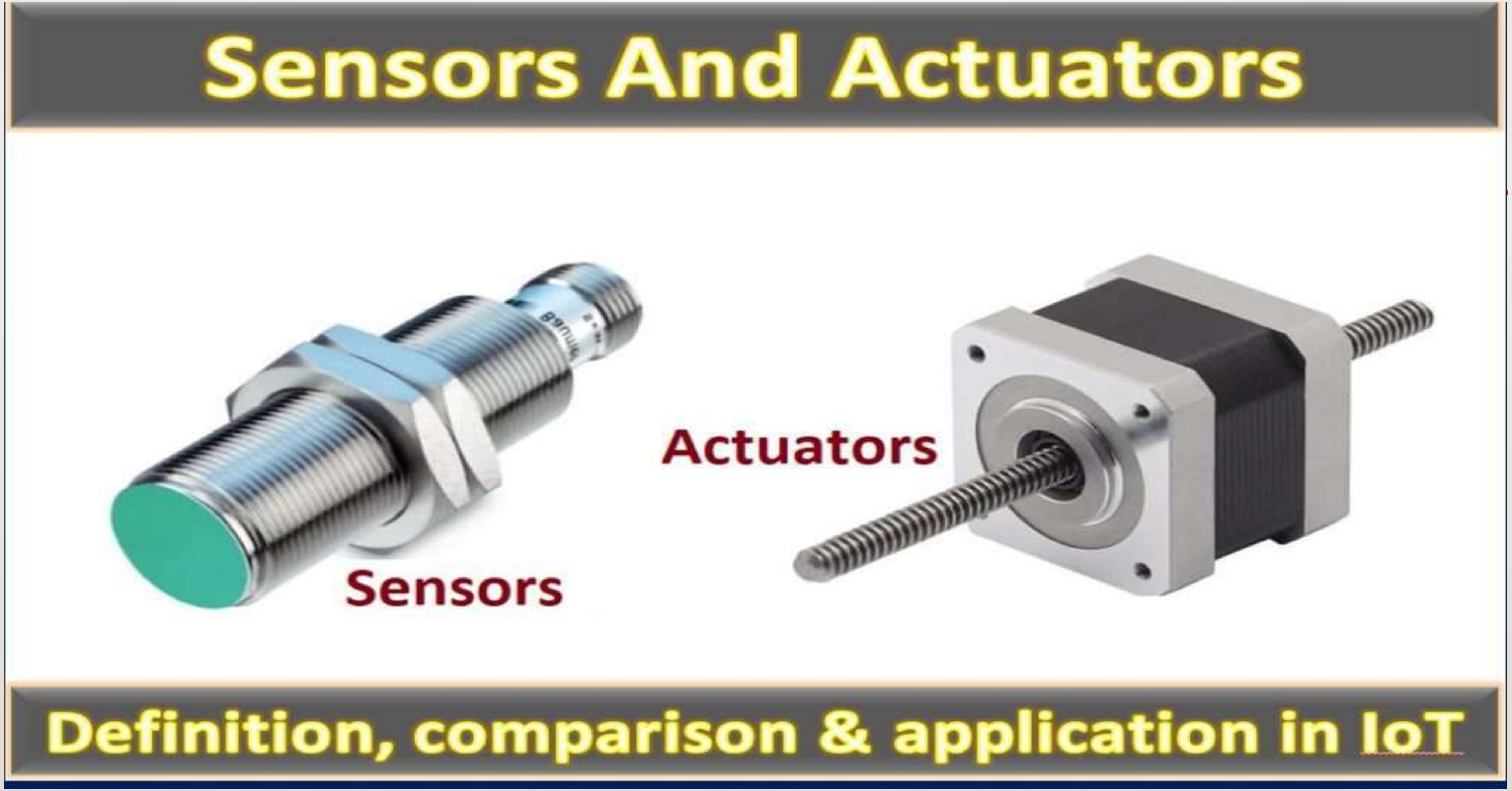
An embedded system is a computer designed to perform dedicated functions within a larger system.

Examples: Microcontrollers in appliances, automotive control systems, medical devices.

Key Components of Embedded Systems:

- Microcontroller/Microprocessor: The brain of the system.
- Memory: Storage for data and instructions.
- Input/Output Interfaces: Communication with the external environment.
- Sensors: Devices that detect changes in the environment.
- Actuators: Devices that perform actions based on sensor input.

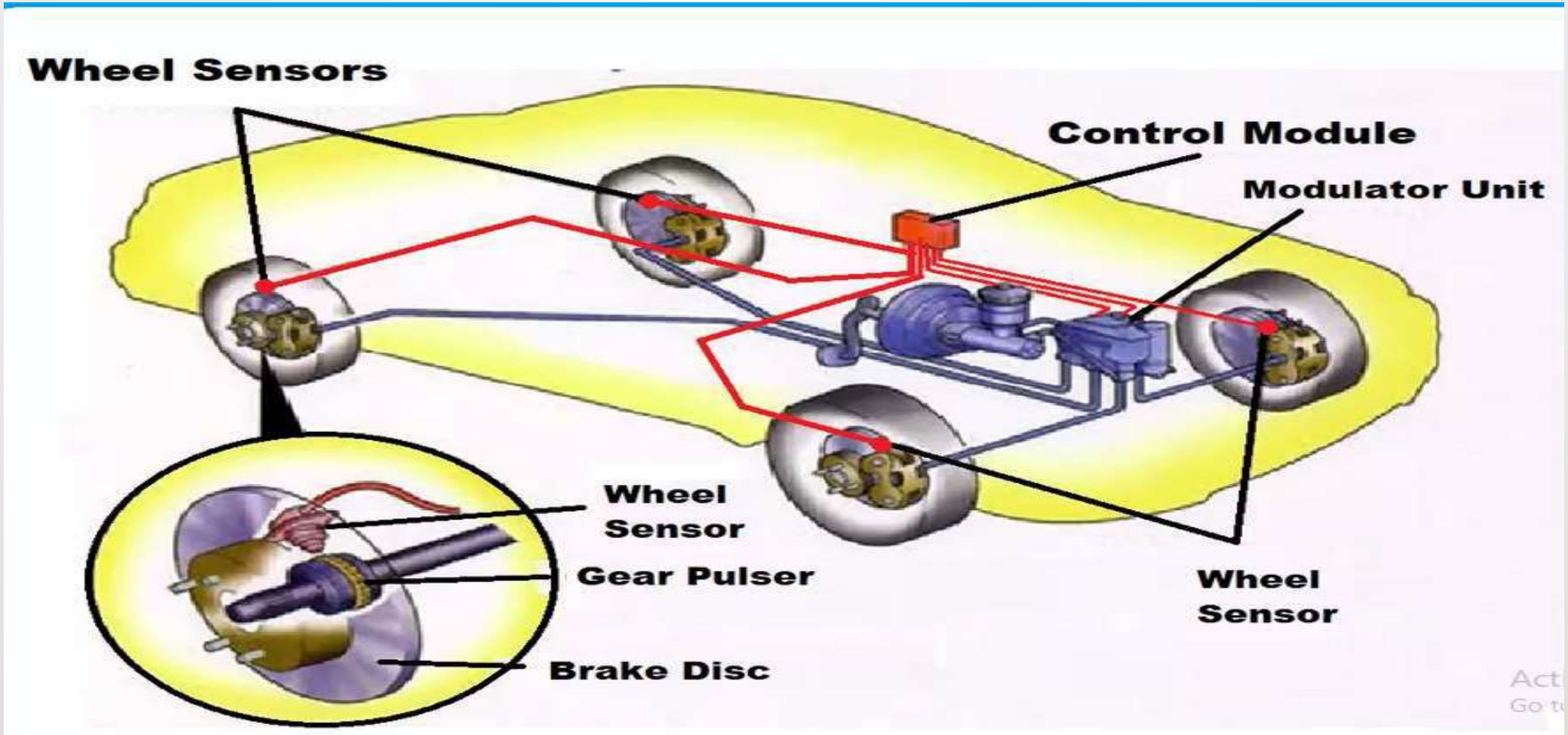

System On Chip
TestController
JTAG/TAP
MBIST
SRAM
CORE LOGIC
Compression Logic for SCAN
DDR
PVT
PLL
SERDES
SERDES
SERDES
BSCAN
BSCAN


Sensors And Actuators
Actuators
Sensors
Definition, comparison & application in IoT

SENSORS

- Sensors are devices that convert physical phenomena (temperature, light, pressure, etc.) into electrical signals.
- Types of Sensors
- Temperature Sensors: Thermocouples, thermistors, and digital temperature sensors (e.g., DS18B20).
- Pressure Sensors: Strain gauges, piezoelectric sensors.
- Light Sensors: Photodiodes, phototransistors, and light-dependent resistors (LDR).
- Motion Sensors: Passive infrared (PIR) sensors, accelerometers.
- Proximity Sensors: Ultrasonic, capacitive, and inductive sensors.

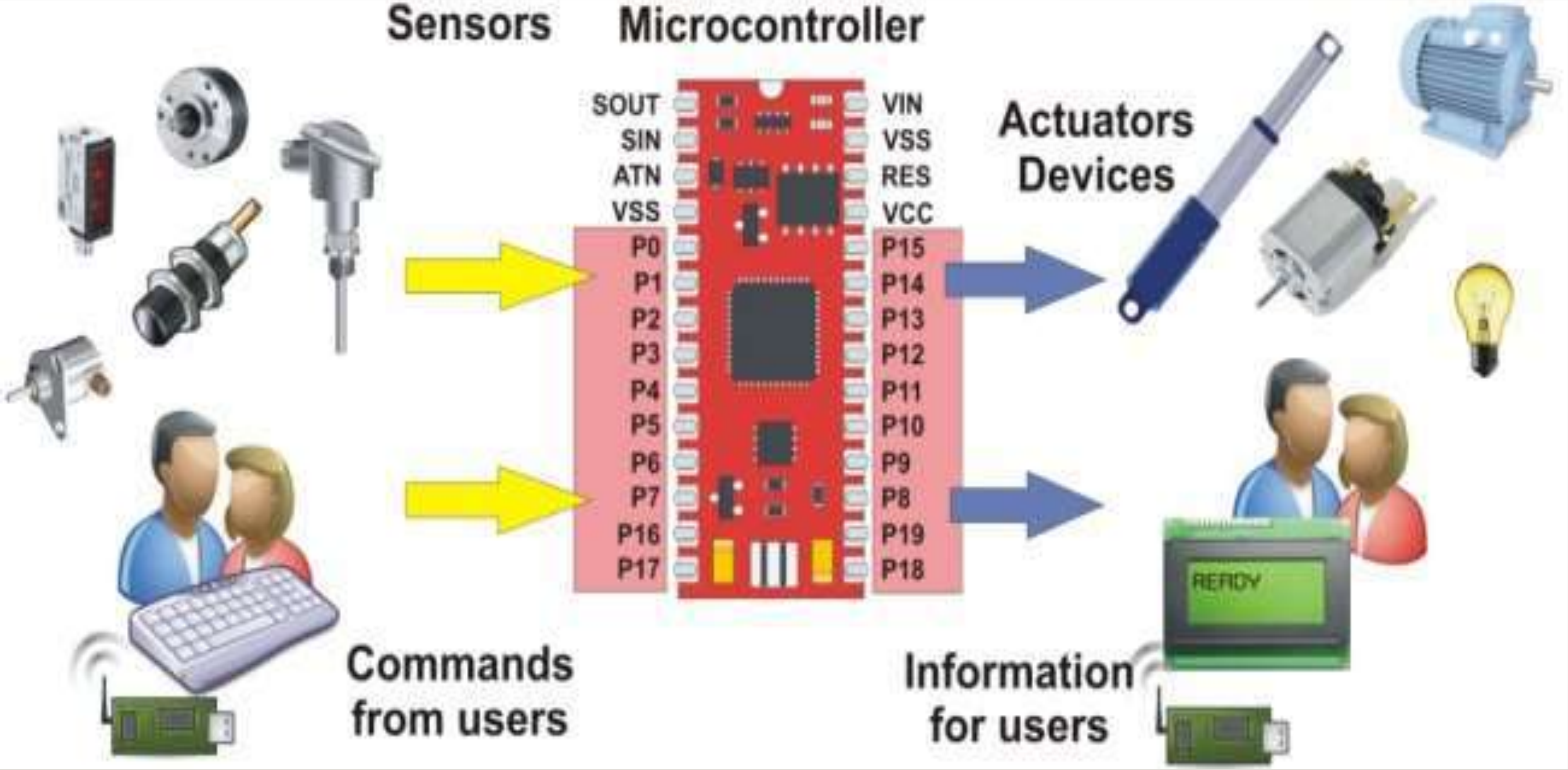
TYPES OF SENSORS


Wheel Sensors
Control Module
Modulator Unit
Wheel
Sensor
Gear Pulser
Brake Disc
Wheel
Sensor

SENSORS

Sensor Characteristics

- Sensitivity: The ability to detect small changes in the measured variable.
- Range: The span of values the sensor can measure.
- Accuracy: The degree to which the sensor's output corresponds to the actual value.
- Response Time: The time taken by the sensor to respond to a change.


Sensors
Microcontroller
SOUT
SIN
ATN
VSS
P0
P1
P2
P3
P4
P5
P6
P7
P16
P17
VIN
VSS
RES
VCC
P15
P14
P13
P12
P11
P10
P9
P8
P19
P18
Actuators
Devices
READY
Commands
from users
Information
for users

ACTUATORS

Actuators are devices that convert electrical signals into physical actions.

- Types of Actuators
- Electric Motors: DC motors, stepper motors, and servo motors.
- Solenoids: Electromagnetic devices that create linear motion.
- Relays: Electrically operated switches.
- Pneumatic Actuators: Use compressed air to create motion.
- Hydraulic Actuators: Use pressurized fluid to create motion.

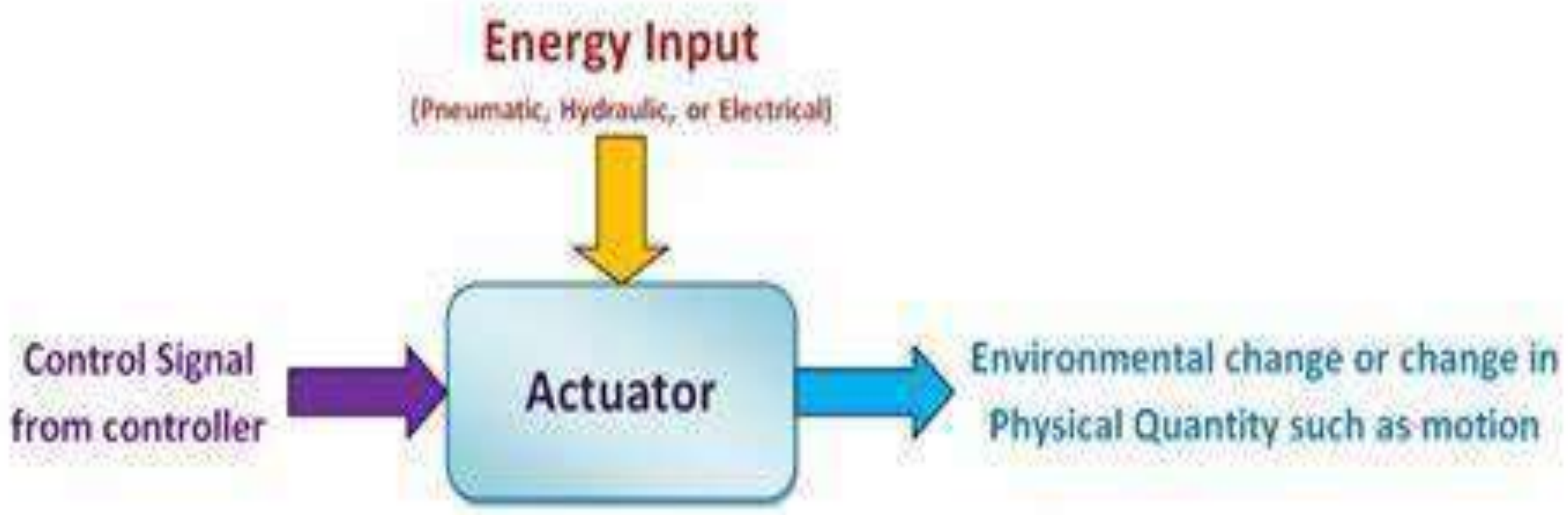


Figure- Block diagram of an actuator

ACTUATORS

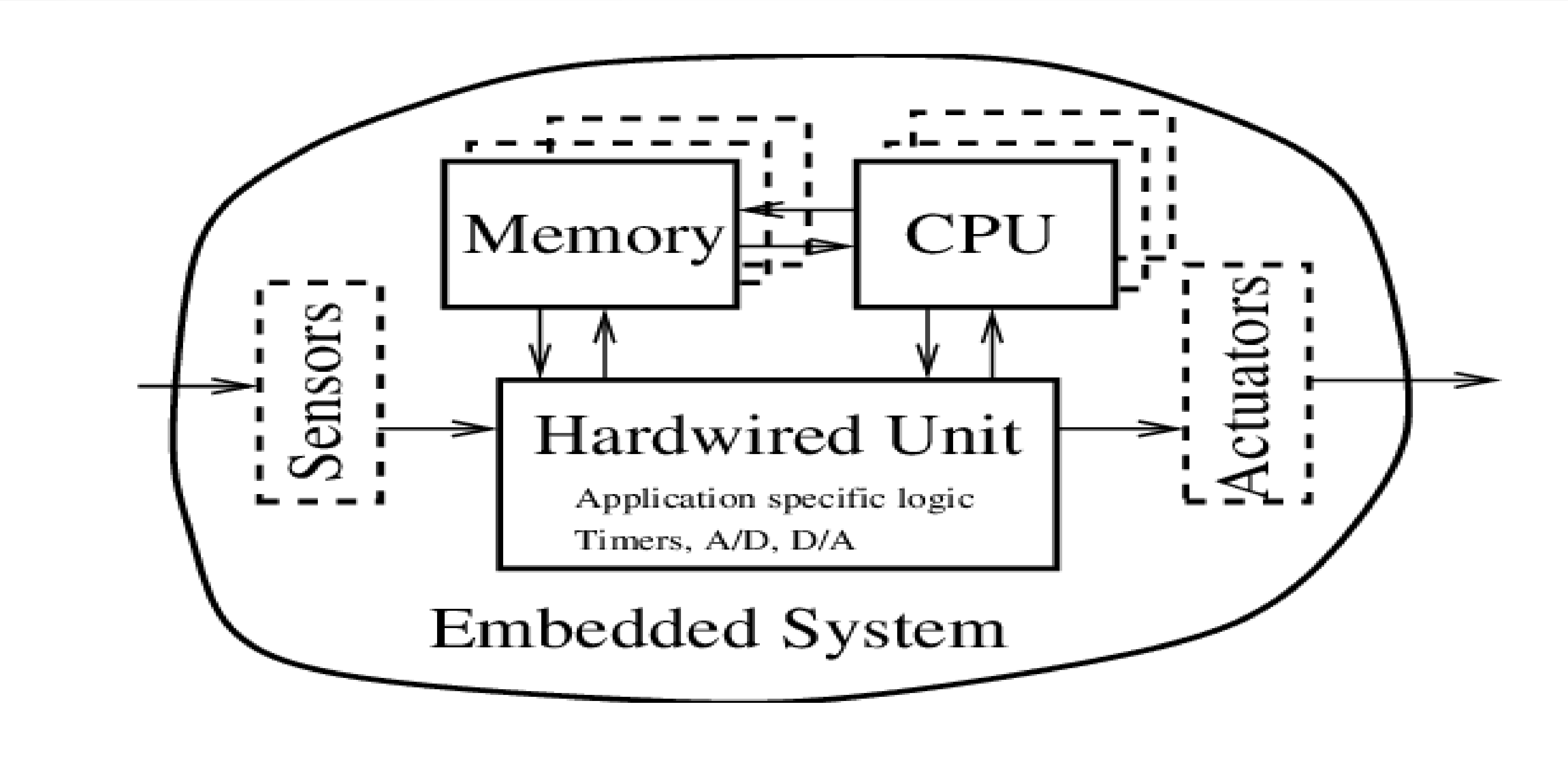
Actuator Characteristics

- Torque: The rotational force produced by motors.
- Speed: The rate at which the actuator can move.
- Power Consumption: The amount of power required to operate the actuator.
- Control Precision: The accuracy with which the actuator can be controlled.

MEMORY IN EMBEDDED SYSTEMS

Types of Memory

- Volatile Memory: Loses data when power is off.
- RAM (Random Access Memory): Used for temporary data storage.
- SRAM (Static RAM): Faster and more expensive, used for cache.
- DRAM (Dynamic RAM): Slower, used for main memory.
- Non-Volatile Memory: Retains data even when power is off.


Memory
CPU
Sensors
Hardwired Unit
Application specific logic
Timers, A/D, D/A
Actuators
Embedded System

MEMORY IN EMBEDDED SYSTEMS

- ROM (Read-Only Memory): Permanent storage for firmware.
- Flash Memory: Rewritable non-volatile memory used for data storage.
- EEPROM (Electrically Erasable Programmable Read-Only Memory): Used for small amounts of data that need to be preserved.
- Memory Considerations in Embedded Design
- Capacity: Amount of data that can be stored.
- Speed: Access time for reading/writing data.
- Endurance: Number of write/erase cycles before failure.
- Cost: Budget constraints for memory components.

INTEGRATION OF SENSORS AND ACTUATORS

- Sensor-Actuator Interaction
- Sensors provide data to the microcontroller, which processes the information and sends commands to actuators.
- Example: A temperature sensor detects high temperature and activates a cooling fan.
- Control Systems
- Open-loop Control: No feedback; the actuator operates based on predefined commands.
- Closed-loop Control: Feedback is used to adjust the actuator's operation based on sensor data.

DESIGN CONSIDERATIONS

- Latency: Time delay between sensor reading and actuator response.
- Power Management: Efficient use of power, especially in battery-operated devices.
- Safety and Reliability: Ensuring the system operates correctly under all conditions.

MEMORY TYPES IN EMBEDDED SYSTEMS

- ROM (Read-Only Memory)
- Characteristics:
- Non-volatile: Retains data without power.
- Typically used to store firmware or boot code.
- Types of ROM:
- PROM (Programmable ROM): Can be programmed once.
- EPROM (Erasable Programmable ROM): Can be erased using UV light and reprogrammed.
- EEPROM (Electrically Erasable Programmable ROM): Can be erased and reprogrammed electrically.
- Flash Memory: A type of EEPROM that allows for faster erase and write cycles.

MEMORY TYPES IN EMBEDDED SYSTEMS

- RAM (Random Access Memory)
- Characteristics:
- Volatile: Loses data when power is off.
- Used for temporary data storage during operation.
- Types of RAM:
- SRAM (Static RAM): Faster, used for cache memory, retains data as long as power is supplied.
- DRAM (Dynamic RAM): Slower, requires periodic refreshing, used for main memory.

MEMORY ACCORDING TO THE TYPE OF INTERFACE

- Parallel Interface
- Data is transferred simultaneously across multiple lines.
- Advantages: Faster data transfer rates.
- Disadvantages: More complex wiring and increased cost.

MEMORY TYPES IN EMBEDDED SYSTEMS

- Serial Interface
- Data is transferred one bit at a time over a single line.
- Advantages: Simpler wiring, lower cost.
- Disadvantages: Slower data transfer rates.
- Examples: SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver-Transmitter).

MEMORY SHADOWING

- :A technique where a copy of data or code is stored in a faster memory (like SRAM) to improve access speed.
- Use Case: Frequently accessed data or critical code segments are shadowed in faster memory to reduce latency.
- Implementation: The system checks the shadow memory first before accessing slower memory.

MEMORY SELECTION FOR EMBEDDED SYSTEMS

Factors to Consider:

- Capacity: Amount of data needed for the application.
- Speed: Access time and data transfer rates.
- Endurance: Number of write/erase cycles (especially for flash memory).
- Cost: Budget constraints for the project.
- Power Consumption: Important for battery-operated devices.
- Form Factor: Size and physical layout of the memory chip.

SENSORS AND ACTUATORS AS I/O COMPONENTS

- Definition: Devices that detect physical phenomena and convert them into electrical signals.
- Functionality: Provide input to the embedded system for processing.

TYPES OF SENSORS

- Temperature Sensors: Thermocouples, thermistors.
- Pressure Sensors: Strain gauges, piezoelectric sensors.
- Light Sensors: Photodiodes, LDRs.
- Motion Sensors: Accelerometers, gyroscopes.
- Proximity Sensors: Ultrasonic, capacitive sensors.

SENSOR CHARACTERISTICS

- Sensitivity: Ability to detect small changes.
- Range: Span of measurable values.
- Accuracy: Degree of closeness to the actual value.
- Response Time: Time taken to respond to changes.

OVERVIEW OF ACTUATORS

- Devices that convert electrical signals into physical actions.
- Functionality: Perform actions based on commands from the embedded system.

TYPES OF ACTUATORS

- Electric Motors: DC motors, stepper motors, servo motors.
- Solenoids: Create linear motion.
- Relays: Electrically operated switches.
- Pneumatic Actuators: Use compressed air for motion.
- Hydraulic Actuators: Use pressurized fluid for motion.

ACTUATOR CHARACTERISTICS

- Torque: Rotational force produced.
- Speed: Rate of movement.
- Power Consumption: Energy required for operation.
- Control Precision: Accuracy of control over movement.

RELAYS

- A relay is an electromechanical switch that uses an electromagnetic coil to open or close a circuit. It allows a low-power signal to control a high-power circuit.

Components of a Relay

- Electromagnetic Coil: When energized, it creates a magnetic field.
- Armature: A movable lever that is attracted to the coil when energized.
- Contacts: Metal pieces that open or close the circuit. There are typically two types:
- Normally Open (NO): The circuit is open when the relay is not energized and closes when energized.
- Normally Closed (NC): The circuit is closed when the relay is not energized and opens when energized.

RELAY OPERATION

- When a voltage is applied to the coil, it generates a magnetic field.
- The magnetic field attracts the armature, which moves to make or break the connection between the contacts.
- This allows the relay to control a larger load (like a motor or light) with a smaller control signal.

APPLICATIONS

- Home Automation: Control lights, fans, and appliances.
- Automotive: Control high-current devices like headlights and motors.
- Industrial Automation: Control machinery and equipment

ADVANTAGES AND DISADVANTAGES

Advantages:

- Electrically isolated control circuit from the load circuit.
- Can switch high voltages and currents.

Disadvantages:

- Mechanical wear over time.
- Slower switching speeds compared to solid-state devices.

MOTORS WITH PWM CONTROL

- Pulse Width Modulation (PWM) is a technique used to control the amount of power delivered to electrical devices by varying the width of the pulses in a pulse train.
- PWM works by turning the power on and off at a high frequency. The ratio of the "on" time to the total cycle time is called the duty cycle.
- Duty Cycle:
- 0%: Off
- 50%: On half the time, resulting in half the power.
- 100%: Fully on.

APPLICATIONS OF PWM IN MOTORS

- Speed Control: By adjusting the duty cycle, the speed of DC motors can be controlled. A higher duty cycle results in higher speed.
- Torque Control: In stepper motors, PWM can be used to control the current supplied to the motor coils, affecting torque.

ADVANTAGES OF PWM CONTROL

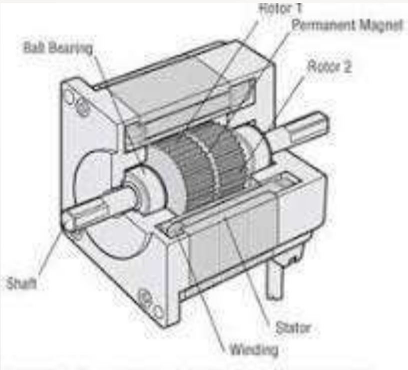
- Efficiency: Reduces power loss compared to linear control methods.
- Precision: Allows for fine control of motor speed and torque.
- Heat Management: Generates less heat in the control circuit.

EXAMPLE OF PWM CONTROL IN DC MOTORS

- A microcontroller generates a PWM signal that drives a transistor or MOSFET, which in turn controls the power to the motor. By changing the duty cycle of the PWM signal, the speed of the motor can be adjusted.

STEPPER MOTORS

- A stepper motor is a type of DC motor that divides a full rotation into a number of equal steps. It moves in discrete steps, allowing for precise control of position and speed.



Motor Structural Diagram: Cross-Section Parallel to Shaft

TYPES OF STEPPER MOTORS

- Permanent Magnet Stepper: Uses permanent magnets in the rotor. Provides good torque at low speeds.
- Variable Reluctance Stepper: Relies on the magnetic reluctance principle. Generally has lower torque.
- Hybrid Stepper: Combines features of both permanent magnet and variable reluctance types. Offers high performance and precision.

OPERATION

- Stepper motors are controlled by energizing the coils in a specific sequence. Each pulse sent to the motor corresponds to a step, allowing for precise control of the motor's position.
- Full Step: The motor moves one full step with each pulse.
- Half Step: The motor alternates between full steps and half steps, effectively doubling the resolution.

CONTROL METHODS

- Unipolar Control: Uses a single coil per phase, allowing for simpler control but less torque.
- Bipolar Control: Uses two coils per phase, allowing for higher torque but requiring more complex control circuitry.

APPLICATIONS

- 3D Printers: Precise control of the print head and bed.
- Robotics: Control of joint positions and movements.

TIMERS

- A timer is a hardware component that counts time intervals and can generate interrupts or trigger events after a specified duration.
- Timers are essential for scheduling tasks, generating delays, and measuring time intervals in embedded systems.

TYPES OF TIMERS

- One-Shot Timer: Generates a single pulse after a specified time period.
- Periodic Timer: Generates pulses at regular intervals.
- Real-Time Timer: Maintains accurate time and can be used for timekeeping applications.

APPLICATIONS OF TIMERS

- Delay Generation: Creating time delays for various operations.
- Event Scheduling: Triggering events at specific intervals.
- PWM Generation: Controlling the duty cycle of PWM signals.

TIMER CONFIGURATION

- Prescaler: Divides the clock frequency to extend the timer's range.
- Counter Register: Holds the current count value.
- Compare Register: Sets the value at which an interrupt is generated.

COUNTERS

- A counter is a digital device that counts the number of occurrences of an event, such as clock pulses or external signals.
- Counters can be used for frequency measurement, event counting, and timekeeping.

TYPES OF COUNTERS

- Asynchronous Counter: Each flip-flop is triggered by the previous one, leading to propagation delays.
- Synchronous Counter: All flip-flops are triggered simultaneously by a common clock signal, providing faster operation.

APPLICATIONS OF COUNTERS

- Event Counting: Counting the number of occurrences of an event (e.g., button presses).
- Frequency Measurement: Measuring the frequency of an input signal.
- Time Measurement: Counting clock pulses to measure time intervals.

COUNTER CONFIGURATION

- Up Counter: Counts upwards from 0 to a maximum value.
- Down Counter: Counts downwards from a maximum value to 0.
- Up/Down Counter: Can count in both directions based on control signals.

RESET CIRCUIT

- A reset circuit initializes the microcontroller or system to a known state upon power-up or when a reset signal is received.
- It ensures that the system starts in a predictable state.

COMPONENTS OF A RESET CIRCUIT

- Reset Button: Manually triggers a reset.
- Reset IC: Monitors power supply levels and generates a reset signal.
- Capacitor: Creates a delay to ensure stable power before releasing the reset.

APPLICATIONS OF RESET CIRCUITS

- Power-On Reset: Ensures the system starts correctly after power is applied.
- Manual Reset: Allows users to reset the system during operation.

BROWN-OUT PROTECTION CIRCUIT

- Brown-out protection prevents the microcontroller from operating under low voltage conditions, which can lead to unpredictable behavior.

HOW IT WORKS

- The circuit monitors the supply voltage and generates a reset signal if the voltage drops below a predefined threshold.
- This ensures that the system is reset and does not attempt to operate under insufficient voltage conditions.

APPLICATIONS

- Microcontrollers: Protects against brown-out conditions that can cause data corruption.
- Battery-Powered Devices: Ensures reliable operation as battery voltage drops.

OSCILLATOR UNIT

- An oscillator is a circuit that generates a periodic signal, typically a square wave or sine wave, used to provide clock signals for digital circuits.

TYPES OF OSCILLATORS

- Crystal Oscillator: Uses a quartz crystal to provide high stability and accuracy.
- RC Oscillator: Uses resistors and capacitors to generate oscillations, typically less stable than crystal oscillators.
- LC Oscillator: Uses inductors and capacitors, often used in RF applications.

APPLICATIONS OF OSCILLATORS

- Clock Generation: Provides clock signals for microcontrollers and digital circuits.
- Signal Generation: Used in communication systems and audio applications.

REAL-TIME CLOCK (RTC)

- A real-time clock is a specialized timer that keeps track of the current time and date
- A Real-Time Clock (RTC) is a specialized timekeeping device that keeps track of the current time and date, even when the main power supply to the system is turned off. RTCs are commonly used in embedded systems to maintain accurate time and date information.

KEY FEATURES OF RTCS

- Battery Backup: RTCs typically have a small battery (like a coin cell) that allows them to keep time even when the main power is off.
- Low Power Consumption: Designed to consume minimal power, allowing them to run for extended periods on battery power.
- Timekeeping Accuracy: RTCs are designed to maintain accurate time, often with a drift of only a few seconds per month.

COMPONENTS OF AN RTC

- Oscillator: Usually a crystal oscillator that provides a stable clock signal.
- Counter: A counter that increments based on the oscillator's frequency to keep track of seconds, minutes, hours, days, etc.
- Interface: Communication interface (like I2C or SPI) to allow the microcontroller to read and set the time.

APPLICATIONS OF RTCS

- Embedded Systems: Used in devices that require accurate timekeeping, such as data loggers, alarms, and timers.
- Consumer Electronics: Found in clocks, watches, and home appliances.
- Industrial Applications: Used in systems that require time-stamped data for logging and monitoring.
- Example of RTC ICs
- DS1307: A popular I2C RTC with a battery backup.
- PCF8563: Another I2C RTC that provides timekeeping and alarm functions.

WATCHDOG TIMER

- A Watchdog Timer (WDT) is a hardware or software timer that is used to detect and recover from malfunctions in embedded systems. It ensures that the system is functioning correctly by monitoring the execution of the program.

HOW WATCHDOG TIMERS WORK

- The WDT is set to a specific timeout period. The main program must periodically reset (or "kick") the watchdog timer within this period.
- If the program fails to reset the timer (due to a crash, infinite loop, or other malfunction), the WDT will expire and trigger a system reset or an interrupt.

KEY FEATURES OF WATCHDOG TIMERS

- Timeout Period: Configurable duration after which the WDT will trigger a reset if not reset by the program.
- Reset Mechanism: Can either reset the entire system or generate an interrupt for further handling.
- Independent Operation: Often operates independently of the main program, ensuring that it can detect failures even if the main program is unresponsive.

APPLICATIONS OF WATCHDOG TIMERS

- Embedded Systems: Used in microcontrollers and processors to ensure system reliability.
- Safety-Critical Applications: Common in automotive, medical, and industrial systems where failure can have serious consequences.
- Remote Systems: Useful in devices that are difficult to access physically, allowing for automatic recovery from faults.

EXAMPLE OF WATCHDOG TIMER IMPLEMENTATION

- In a microcontroller, the WDT is initialized with a specific timeout value. The main loop of the program includes a call to reset the WDT at regular intervals. If the program hangs or crashes, the WDT will not be reset, leading to a system reset.

CASE STUDIES AND APPLICATIONS

- Real-World Applications
- Home Automation: Smart thermostats, lighting control systems.
- Automotive Systems: Engine control units, anti-lock braking systems.
- Medical Devices: Heart rate monitors


Thank you