



Physical Computing HS25

What is a Sensor?

A sensor is a translator: it converts a **physical phenomenon** into an **electrical signal** that a microcontroller can read.

World → Sensor → Signal → Arduino → Response

What is a Sensor?

Type	What it gives you	Examples
Digital	ON / OFF (1 or 0)	Button, motion detector (PIR), tilt switch, touch sensor
Analog	Range of values (0–1023)	Potentiometer, light sensor (LDR), pressure sensor, temperature

Some sensors use protocols (I2C, SPI) to send more complex data – we've covered this.

Sensor Categories by What They Sense

Category	Senses...	Examples
Presence / Proximity	Is something/someone there? How close?	PIR, ultrasonic, IR distance, ToF, capacitive touch
Light	Brightness, color, specific wavelengths	LDR, phototransistor, UV sensor, color sensor
Sound	Volume, frequency, specific sounds	Microphone, sound level sensor
Touch / Pressure	Contact, force, stretch	FSR, piezo, capacitive, flex sensor
Motion / Orientation	Movement, tilt, rotation, direction	Accelerometer, gyroscope, IMU, compass
Environment	Temperature, humidity, air quality	Thermistor, DHT11/22, gas sensors, CO2
Distance	How far away is something?	Ultrasonic (HC-SR04), IR, ToF (VL53L0X), LIDAR

Choosing a Sensor – Questions to Ask

1. What do you actually want to know?
 - Not "I need a motion sensor" but "I want to know if someone is approaching"
2. What range / precision do you need?
 - Detecting presence in a room \neq measuring exact distance in mm
3. What are the conditions?
 - Indoors/outdoors? Lighting? Noise? Weather?
4. How fast does it need to respond?
 - Human reaction time (~200ms) vs. real-time tracking
5. What's your budget / complexity tolerance?
 - Simple analog sensor vs. I2C module vs. camera + ML

Matching Intent to Sensor

If you want to detect...	Consider...
Someone entering a space	PIR, ultrasonic, IR break-beam
How close someone is	Ultrasonic, ToF, IR distance
Touch / contact	Capacitive, FSR, piezo
Gesture / movement direction	Ultrasonic array, accelerometer, camera
Ambient conditions	Light (LDR), temp (DHT), sound (mic)
Something specific (face, object, voice)	Camera + ML, specific modules

Start Simple, Get Specific

Good prototyping pattern:

1. Start with the **simplest sensor** that might work
2. Test it in **real conditions** (not just on your desk)
3. Discover the **limitations**
4. Upgrade to more specific/precise sensor **if needed**

Example: Start with PIR for presence → discover it's too slow → switch to ultrasonic for distance threshold