Integrating Sensors with Robotic Systems

DASE7503 Robotic Systems Integration

Contents

- Introduction
- Sensor Categories
- Sensor Interfaces and Integration
- Common Sensors for Robotics Applications

Introduction

- Be a vast number of different sensors being used in robotics
- Classify sensors by their interface and select a number of typical sensor systems to discuss their specifics in hardware and software
- Be always important to find the right sensor for a particular application, including the right measurement technique, the right size and weight, the right operating temperature range and power consumption, and of course the right price.
- Data transfer from the sensor to the CPU can be either **CPU-initiated (polling)** or **sensor-initiated (via interrupt)**.
  - In case it is CPU-initiated, the CPU has to keep checking whether the sensor is ready by reading a status line in a loop. This is much more time-consuming than the alternative of a sensor-initiated data transfer.
  - The sensor-initiated requires the availability of an interrupt line. The sensor then signals that data is ready via an interrupt, and the CPU can react immediately to this request.



Sensor Categories

- From an engineering point of view, it makes sense to classify sensors according to their output signals

Sensor output	Typical sensor
Binary signal (0 or 1)	Tactile sensor
Analog signal (e.g., 0 … 5 V)	Inclinometer
Timing signal (e.g., PWM)	Gyroscope
Serial link (RS232 or USB)	GPS, IMU
Parallel link	Digital camera

a summary of sensor outputs together with their typical sensors

Sensor Categories

i) From a robot's point of view, it is more important to distinguish:

- **Local** sensors

sensors mounted on the robot.

- **Global** sensors

sensors mounted outside the robot in its environment and transmitting sensor data back to the robot.

ii) For mobile robot systems, it is also important to distinguish:

- **Internal** (proprioceptive) sensors

sensors monitoring the robot's internal state.

- **External** sensors

sensors monitoring the robot's environment.

	Local	**Global**
Internal	Passive Battery sensor Chip-temperature sensor Shaft encoders Accelerometer Gyroscope Inclinometer Compass Active –	Passive – Active –
	Local	**Global**
External	Passive On-board camera Active Sonar sensor Infrared distance sensor Laser scanner	Passive Overhead camera Satellite GPS Active Sonar (or other) Global Positioning System

Sensor classification based on application

Sensor Categories

iii) A further distinction is between:

- **Passive** sensors

sensors that monitor the environment without disturbing it, for example digital camera, gyroscope

- **Active** sensors

sensors that stimulate the environment for their measurement, for example sonar sensor, Lidar laser scanner, infrared sensor.

	Local	**Global**
Internal	Passive Battery sensor Chip-temperature sensor Shaft encoders Accelerometer Gyroscope Inclinometer Compass	Passive –
	Active –	Active –
	Local	**Global**
External	Passive On-board camera	Passive Overhead camera Satellite GPS
	Active Sonar sensor Infrared distance sensor Laser scanner	Active Sonar (or other) Global Positioning System

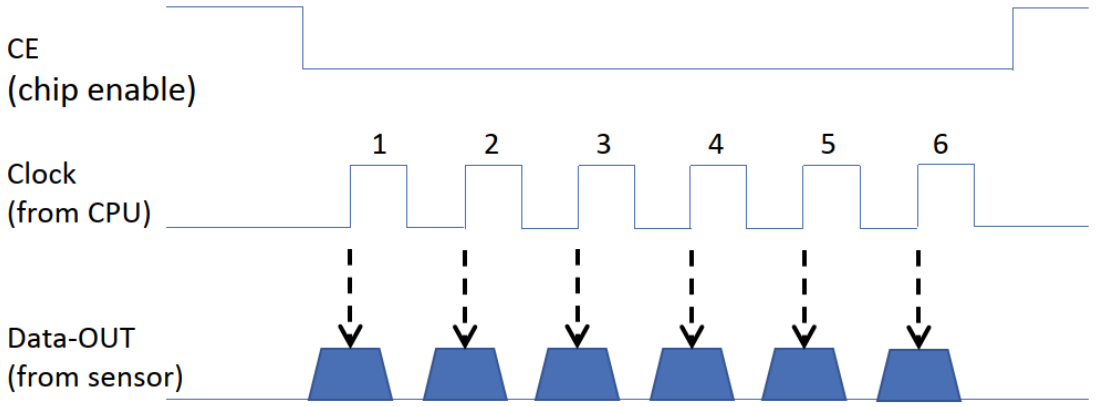
Sensor classification based on application

Synchronous Serial and I²C Interfaces

- The **output signal** of digital sensors can have different forms. It can be a **parallel interface** (e.g., 8 or 16 digital output lines), a **serial interface** (e.g., following the RS232 or USB standard) or a **synchronous serial interface**.

1) Synchronous Serial Interface

- The expression synchronous serial means that the converted data value is **read bit by bit from the sensor**.
- After setting the chip-enable line for the sensor, the CPU sends pulses via a digital output pin and at the same time reads one bit of information from the sensor (e.g., read one bit of data on each rising edge of the pulse pin).

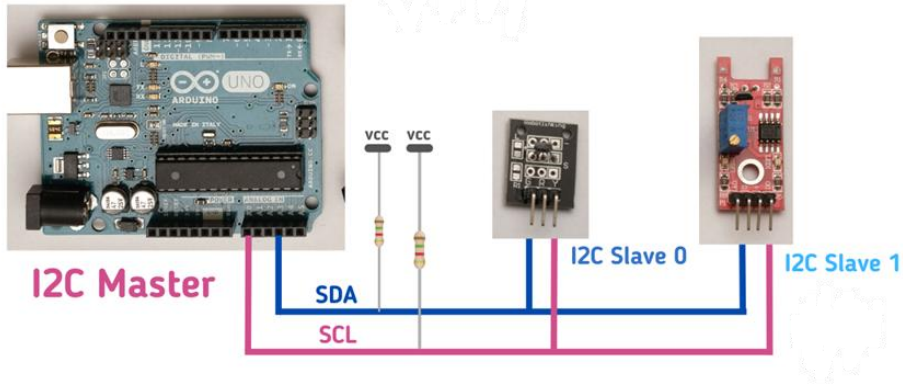


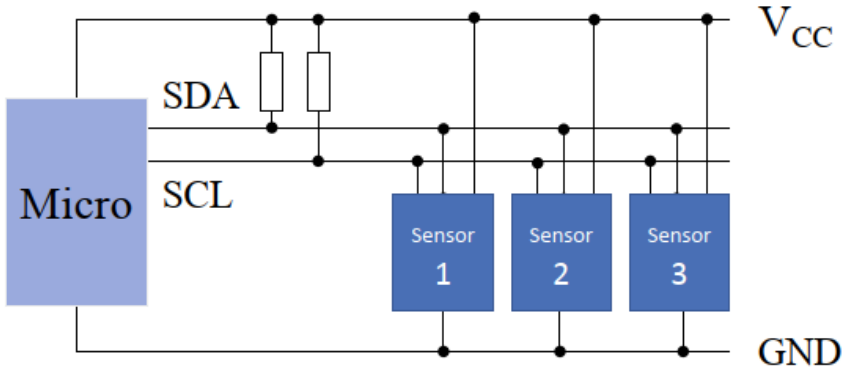
Signal timing for synchronous serial interface

Synchronous Serial and I²C Interfaces

2) I²C Interface

- I²C (Inter-Integrated Circuit, also written as I2C or IIC) is a general-purpose extension of the synchronous serial interface. It allows to link one master module (microprocessor) with several slave modules (e.g., sensors) via **a single port**.
- In addition to supply voltage VCC and ground, the I²C interface only requires two signal lines:
  - SDA serial data line
  - SCL serial clock line
- Both signals have a pull-resistor to VCC (Voltage Common Collector) to indicate that the bus system is idle if none of the slave modules wants to transmit any data.



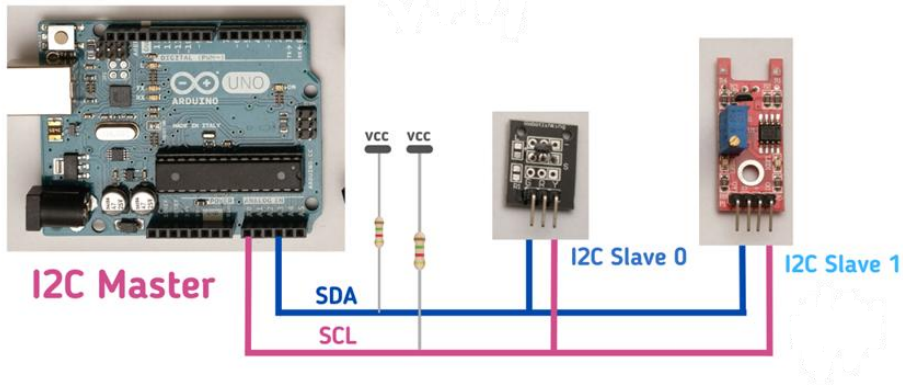


I²C connection schema

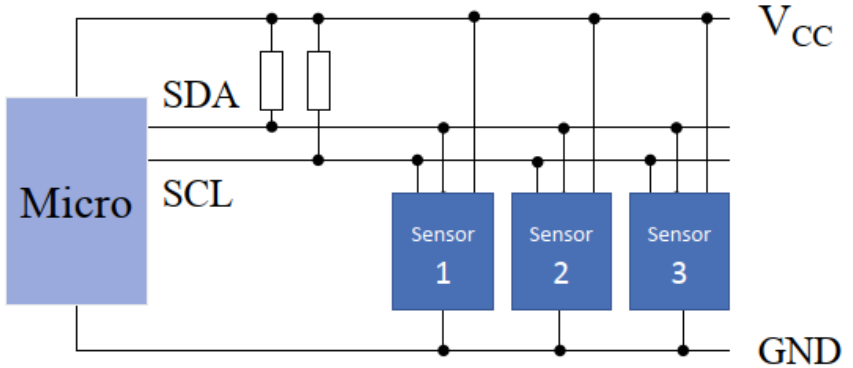
Synchronous Serial and I²C Interfaces

2) I²C Interface

- The master module (in our case the CPU) generates the clock signal for either transmitting or receiving bitwise data via serial data line.
- In the standard I²C protocol, the master transmits in sequence:
  - START signal
  - 7-bit slave address
  - read/write bit (‘0’ for write, ‘1’ for read from the CPU’s point of view).



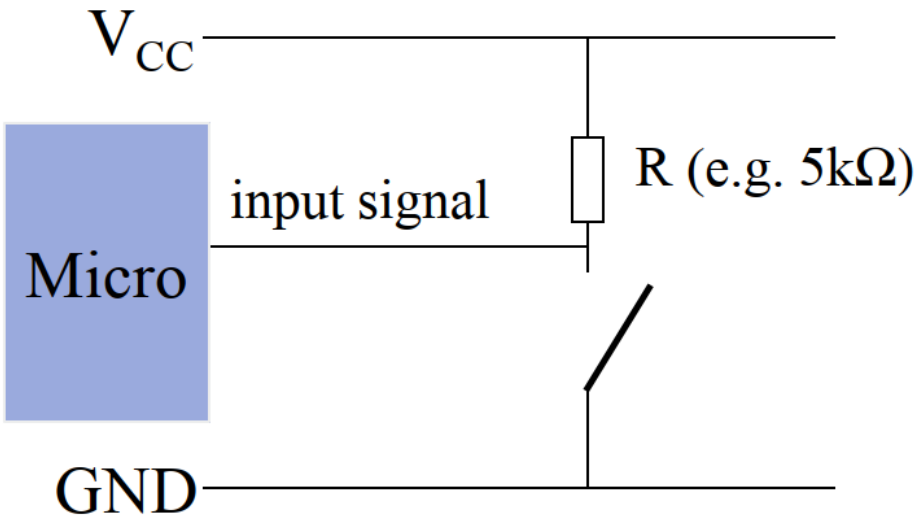
- All data bits are transmitted via the SDA data line, while the CPU maintains timing via the SCL clock line. Communication is terminated by the CPU sending a STOP signal.
- All slave modules need to have settable and unique address values for this mechanism to work. The roles between master and slave can even be swapped, e.g., when connecting several microprocessors via I²C.



I²C connection schema

Sensors – Binary Sensors

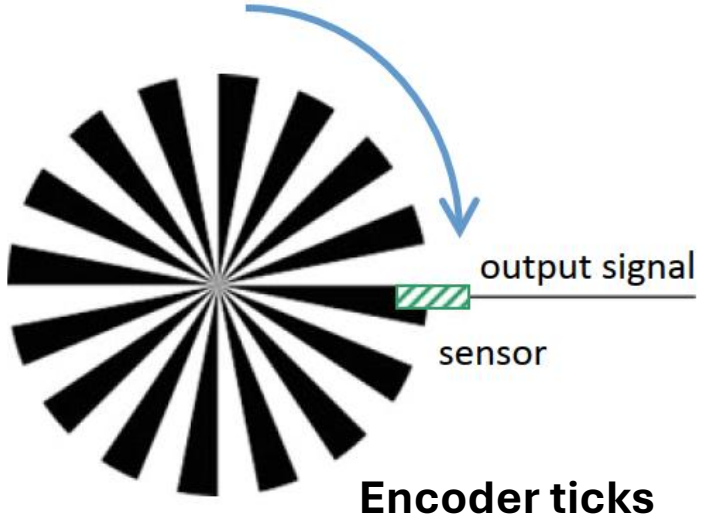
- The simplest type of sensors
- Return a single bit of information, either a 0 or a 1.
- A typical example is a **tactile** sensor on a robot, for example using a microswitch.
- **Interfacing** to a microcontroller can be achieved very easily by using a digital input on the controller.
- How to use a resistor to link to a digital input.
  - A **pull-up resistor** will generate a high signal unless the switch is activated. This is called an **active low setting**.
- A number of microcontrollers, such as the Arduino Nano, have built-in **pull- up and pull-down resistors** on their I/O lines.
- These can be activated via software and simplify the general system design as separate resistors are no longer required.



Interfacing a tactile sensor

Sensors – Shaft Encoders

- Required as fundamental feedback sensors for motor control
- Be several techniques for building an encoder. The most widely used ones are either **magnetic encoders** or **optical encoders**.
- Magnetic encoders use a Hall-effect sensor and a rotating disk on the motor shaft with a number of magnets (e.g., 16) mounted in a circle.
- Every revolution of the motor shaft drives the magnets past the Hall-effect sensor and therefore results in 16 pulses or ticks on the encoder line. The faster the encoder disk spins, the shorter and more frequent will the ticks be.

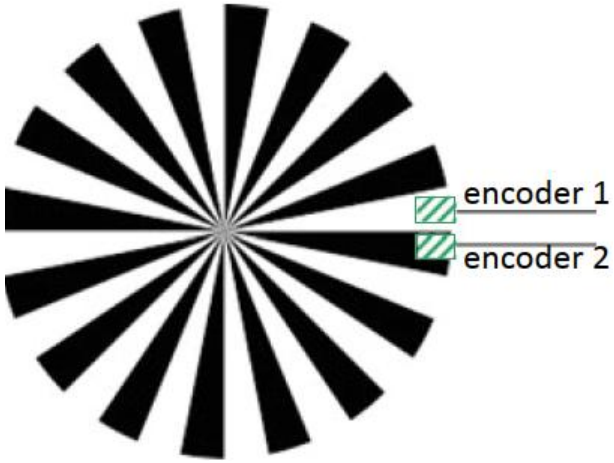


Signal for encoder disk spinning

- Slow:
- Fast:
- Not at all:

Sensors – Shaft Encoders

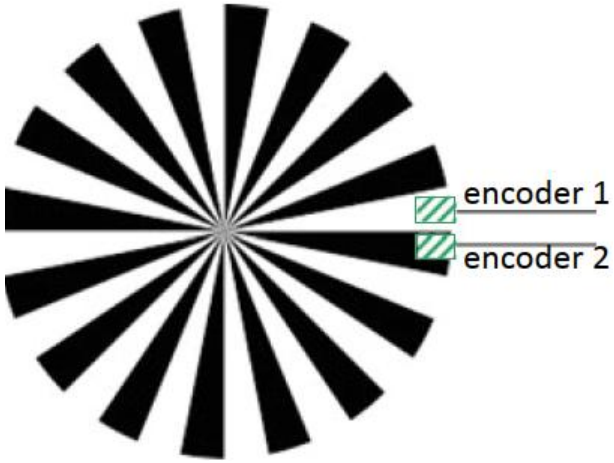
- **Standard optical encoders** use a sector disk with **black and white segments** together with an LED and a photodiode.
- The photodiode detects reflected light during a white segment, but not during a black segment. So, if this disk has 16 white and 16 black segments, the sensor will generate 16 rectangle pulses during one full revolution.
- As has been mentioned above, an **incremental encoder** with only a ***single magnetic*** or ***optical sensor*** element can only count the number of segments passing by.
- But it cannot distinguish whether the motor shaft is moving **clockwise** or **counterclockwise**. This is especially important for applications such as robot vehicles which should be able to **move forward** and **backward**.



Incremental encoder

Sensors – Shaft Encoders

- For this reason, most **incremental encoders** are equipped with **two sensors (magnetic or optical)** that are positioned with a **small phase shift** to each other.
- With this arrangement, it is possible to determine the rotation direction of the motor shaft, since it is recorded which of the two sensors first receives the pulse for a new segment.
- Referring to the Figure, if Enc1 receives the signal first, then the motion is clockwise; if Enc2 receives the signal first, then the motion is counterclockwise.



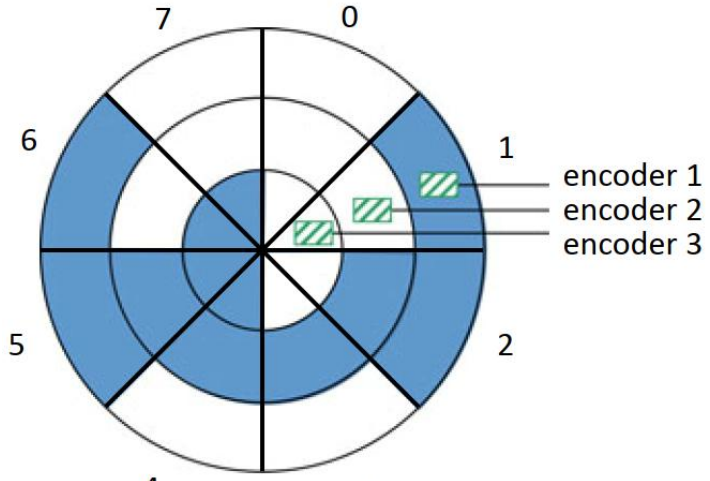
Incremental encoder

Sensors – Shaft Encoders

- Since each of the encoder sensors is just a **binary digital sensor**, we could interface them to a microcontroller by using two digital input lines.
- However, this would not be very efficient, since then the controller would have to constantly poll the sensor data lines in order to record any changes and update the sector count.
- Luckily, this is not necessary, since many modern microcontrollers have special hardware built-in for cases like this. They are called pulse counting registers and can count incoming pulses up to a certain frequency completely independent of the CPU.
- This means the CPU is not being slowed down and is therefore free to work on higher-level application programs. Shaft encoders are standard sensors on mobile robots for determining their position, speed and orientation.

Sensors – Shaft Encoders

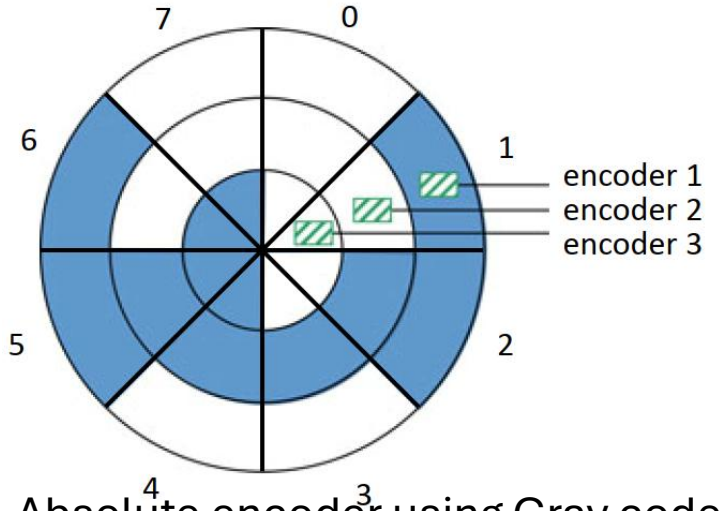
- All encoders described above are called incremental encoders, because they can only count the number of segments passed from a certain starting point.
- They are not sufficient to locate the absolute position of the motor shaft.
- A Gray-code disk can be used in combination with a set of sensors that can solve this issue.
- The **number of sensors** determines **the maximum resolution** of this encoder type (in the example, there are three sensors, giving a resolution of $2^3 = 8$ sectors)



Absolute encoder using Gray code

Sensors – Shaft Encoders

- Note that for any transition between two neighboring sectors of the Gray-code disk, only a single bit changes (e.g., between 1 = 001 and 2 = 011 only the middle bit changes).
- This would not be the case for a standard binary encoding (e.g., 1 = 001 and 2 = 010, which differ by two bits).
- This is an essential feature of this encoder type, because it will still give a proper reading if the disk just passes between two segments.
- For binary encoding the result would be arbitrary when passing between 111 and 000, as it is uncertain which of the three bits would change first, so all bit patterns are possible intermediate outcomes.



Absolute encoder using Gray code

A/D Converters

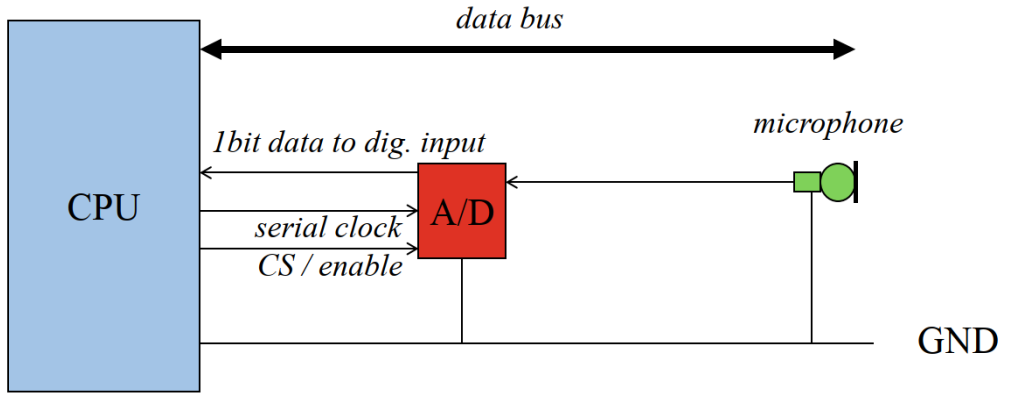
- A number of sensors produce analog output signals rather than digital signals.
- This means an analog-to-digital converter (A/D) is required to connect such a sensor to a microcontroller.
- Typical examples of such sensors are:
  - Microphone
  - Infrared distance sensor
  - Compass
  - Barometer sensor

A/D Converters

- An analog to digital (A/D) is required to convert an analog signal into a digital value.
- The characteristics of an A/D converter include:
  - Accuracy
    - expressed in the number of digits it produces per value, e.g., 10-bit A/D converter
  - Speed
    - expressed in maximum conversions per second, e.g., 500 conversions per
    - second
  - Measurement range
    - expressed in volts, e.g., 0 V to 5 V.

A/D Converters

- The A/D converters come in many variations, and their output format also varies.
- Typical are either a **parallel interface** (e.g., up to 8 bits of accuracy) or a **synchronous serial interface**.
- The **synchronous serial interface** has the advantage that it does not impose any limitations on the number of bits per measurement, for example 10 or 12 bits of accuracy.
- A typical arrangement of an A/D converter interfaced to a CPU as shown in Figure.
- Many A/D converter modules include a multiplexer as well, which allows the connection of several sensors, whose data can be read and converted subsequently.
- In this case, the A/D converter module also has a 1-bit digital data input line, which allows the specification of a particular input line by using a synchronous serial transmission from the CPU to the A/D converter.



A/D converter interfacing

Sensors – Position Sensitive Devices

- Sensors for **distance measurements**, also called position sensitive devices, are among the most important components in robotics.

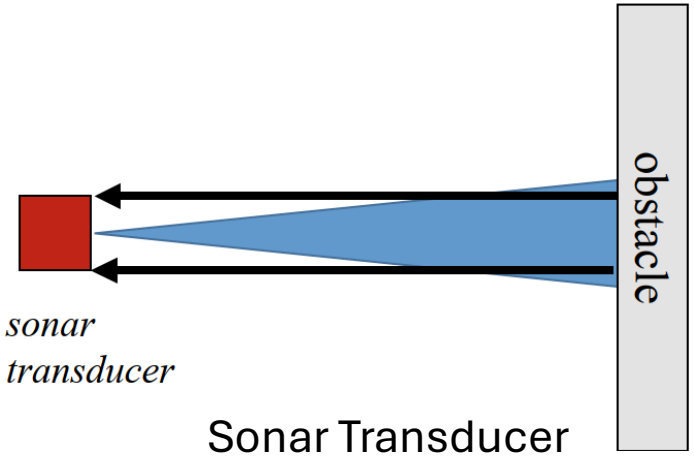
- A few examples like **Sonar**, **Infrared**, and **Laser** sensors

- For decades, mobile robots have been equipped with various sensor types for measuring distances to the nearest obstacle around the robot for navigation purposes.

Sensors – Position Sensitive Devices

Sonar Sensors

- In the past, most robots had been equipped with sonar sensors. Because of the relatively narrow cone of these sensors, a typical configuration to cover the whole circumference of a round robot required 24 sensors, mapping about 15° each.
- Sonar sensors use the principle of a short acoustic signal of about 1 ms at an ultrasonic frequency of 50 kHz to 250 kHz, measuring the time between signal emission until echo reception to the sensor.
- The measured **time-of-flight** is proportional to twice the distance of the nearest obstacle in the sensor cone. If no signal is received within a certain time limit, then no obstacle is detected within the sensor's measuring range.
- Measurements are repeated about 20 times per second, which gives this sensor its typical clicking sound .



Sonar Transducer

Sensors – Position Sensitive Devices

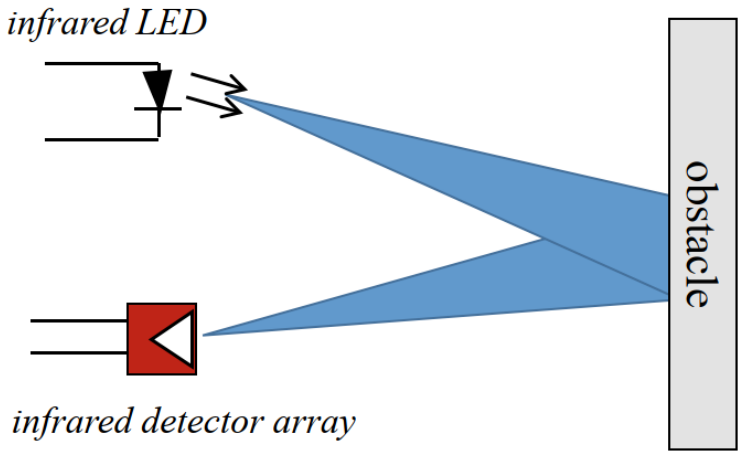
Infrared Sensors

- The **most significant problems** of sonar sensors are inaccuracy, reflections and interference.
- When the acoustic signal is reflected, for example off a wall at a certain angle, then an obstacle seems to be further away than the actual wall that reflected the signal.
- Interference occurs when several sonar sensors are operated at once (among the 24 sensors of one robot or between several robots close to each other).
- Here, it can happen that the acoustic signal from one sensor is being picked up by another sensor, resulting in incorrectly assuming a closer than actual obstacle. Coded sonar signals can be used to prevent this, for example using pseudo- random codes.

Sensors – Position Sensitive Devices

Infrared Sensors

- Infrared (IR) distance sensors do not follow the same principle as sonar sensors, since the time-of-flight for a photon would be much too short to measure with a simple and cheap sensor arrangement.
- Instead, these systems typically use a **pulsed infrared LED** at about 40 kHz together with a detection array as shown in Figure.
- The angle under which the reflected beam is received changes according to the distance to the object and therefore can be used as a measure of the distance.
- The wavelength used is typically 880 nm. Although this is invisible to the human eye, it can be transformed to visible light either by IR detector cards or by recording the light beam with an IR-sensitive camera.

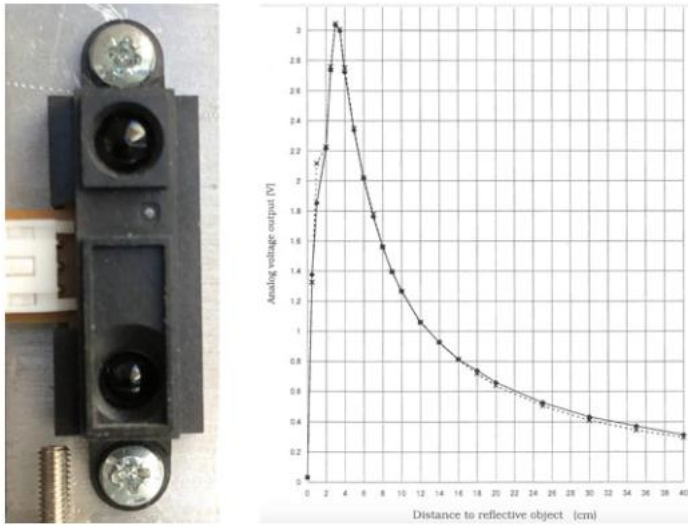


Infrared Sensor

Sensors – Position Sensitive Devices

Infrared Sensors

- The **analog sensor** simply returns a voltage level in relation to the measured distance (unfortunately not proportional, as shown in Figure). The digital sensor has a serial digital interface, transmitting an 8-bit measurement bitwise over a single line, triggered by a clock signal from the CPU
- The relationship between digital sensor read-out (raw data) and actual distance information can be seen. From this diagram, it is clear that the sensor does not return a value linear or proportional to the actual distance, so some post-processing of the raw sensor value is necessary.
- The simplest way of solving this problem is to use a lookup table which can be calibrated for each individual sensor. Since only 8 bits of data are returned, the lookup table will have the reasonable size of 256 entries.
- Another problem becomes evident when looking at the diagram for **actual distances closer than about 6 cm**. These distances are **below the measurement range of this sensor** and will result in an incorrect reading of a much higher distance. This is a serious problem, since it cannot be fixed in a simple way.
- The safest solution is to mechanically mount the sensor in such a way that an obstacle can never get closer than 6 cm or to use an additional **(IR) proximity sensor** to cover for any obstacles closer than this minimum distance.



Sharp PSD sensor and sensor diagram

Sensors - Position Sensitive Devices

Lidar Sensors

- Used in advanced robotics applications as well as in autonomous vehicles, Lidar sensors (light detection and ranging) contain **one or more laser diodes** that point at a rotating mirror, so the sensor can deliver thousands of distance points per second.
- In the future**, solid state Lidars** without any moving parts will take over.



Lidar sensors with single beam

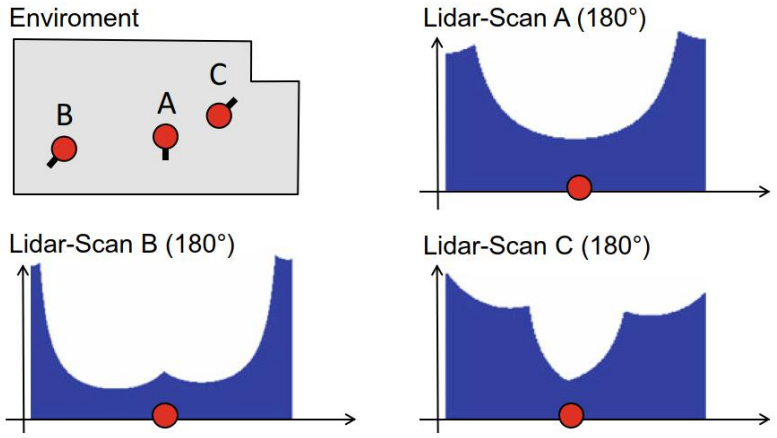
Sensors - Position Sensitive Devices

Lidar Sensors

- Single-beam systems (typically used as industrial safety sensors) generate a single slice through the world.
- Shows three example Lidar scans of 180° each from three different poses A, B and C in the same environment.
- Robot A faces a straight wall, which will translate into a parabola-like shape in the polar coordinates of the Lidar distance measurement.
- Robot B looks directly at a corner, while robot C looks at an inverse corner. This assumes that the Lidar scans progresses clockwise, which is left-to-right from the robot's point of view.



Lidar sensors with single beam



Sample Lidar scans from three different positions and orientations

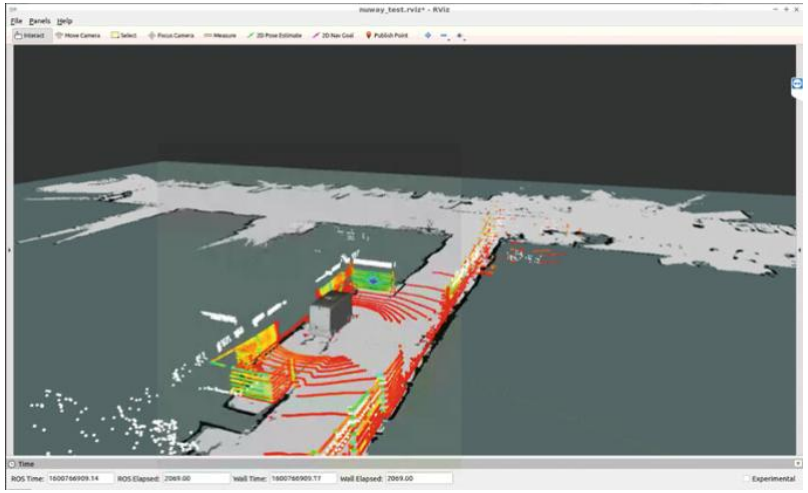
Sensors - Position Sensitive Devices

Lidar Sensors

- **Multi-beam systems** (e.g., Sick Auto Ident[7]) can return an almost perfect local 2D field with distance data, so creating a 3D depth map or point cloud.
- The point cloud generated from multiple Lidar sensors on an autonomous shuttle bus.



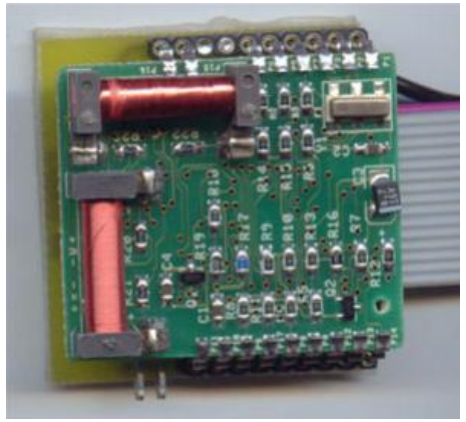
multiple beams



3D point cloud from Lidar sensors

Orientation Sensors

- A **compass sensor** or **magnetometer** is a very useful tool for many mobile robot applications, especially for **localization**, when the robot can rely on its on-board sensors to keep track of its current position and orientation.
- The simplest method for achieving this in a driving robot is to use shaft encoders on each wheel and then apply a method called **dead reckoning**.
- This method starts with a known initial position and orientation (called a pose) and then adds all driving and turning actions to find the robot's current position and orientation.
- Unfortunately, due to **wheel slippage** and other factors, the dead reckoning error will grow larger and larger over time. Therefore, it is a robot's absolute orientation. In many applications, the correct robot orientation is even more important than the correct position.
- A number of magnetic compass sensor modules exist that can be used for work with Arduino or Raspberry Pi microprocessors, e.g., the HMC5883L sensor chip.
- Most modules combine a magnetic compass with three-axis accelerometers and/or gyros to build a more powerful—but also much more expensive—**inertial measurement unit (IMU)**, as explained in the following section.



Vector 2X compass

Inertial Measurement Units

- An inertial measurement unit (IMU) is an internal sensor system for measuring changes in a robot's spatial orientation.
- An IMU typically comprises **three accelerometers** and **three gyroscopes** (one for each axis), sometimes also magnetometers, plus an embedded processor for integrating the sampled data.
- IMU prices range from tens of dollars to tens of thousands of dollars, depending on their sensor and processing quality.
- An example of IMU sensor, which comprises three accelerometer, three gyroscopes and a three-axis magnetic compass and communicates to a controller via its USB interface.



Global Navigation Satellite Systems

- The Global Positioning System (GPS), started by the US Department of Defense in 1973, (civilian use since 1980) was the world's first Global Navigation Satellite System (GNSS).
- Since then, Russia's GLONASS, China's BeiDou and Europe's Galileo have been created as additional independent systems.
- A simple GPS receiver with USB connection (left) and a more sophisticated GPS/IMU combination from Xsens (right).



Global Navigation Satellite Systems

- A GNSS receiver module will automatically connect to several satellites and calculate its own position from their relative data. **Orientation data** is only available from triangulation when a GNSS receiver is in motion.
- When a GPS sensor gets powered up, it continually sends position data, for example in National Marine Electronics Association (NMEA) format.
- NMEA presents the same data in different formats, but we only need to listen to the $GPGGA tokens at the beginning of a line.
- Read out the GPS coordinates shown in blue color:

```
3152.6044,S,11554.2536,E
```

Global Navigation Satellite Systems

Address

999077 China, Hong Kong, Shek Tong Tsui, Sui

Get GPS Coordinates

DD (decimal degrees)*

Latitude 22.2837263236207

Longitude 114.13543750647673

Get Address

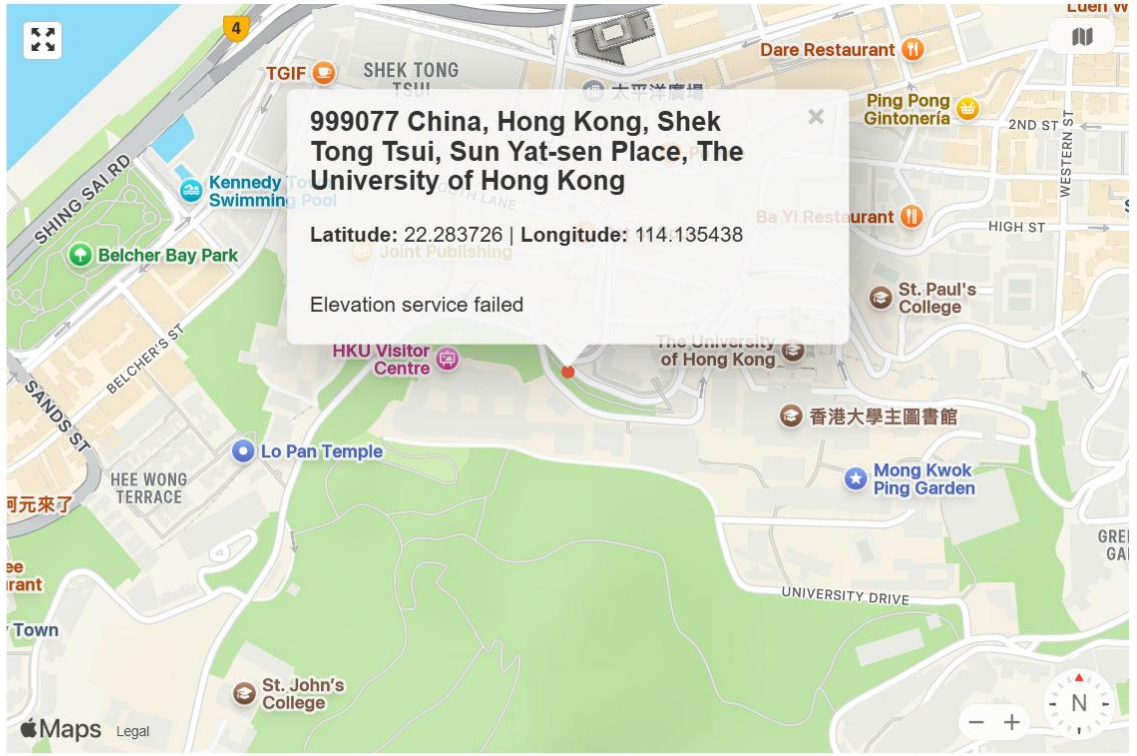
Lat,Long 22.2837263236207,114.135437506

DMS (degrees, minutes, seconds)*

Latitude ◉ N ○ S 22 ° 17 ' 1.415 "

Longitude ◉ E ○ W 114 ° 8 ' 7.575 "

Get Address



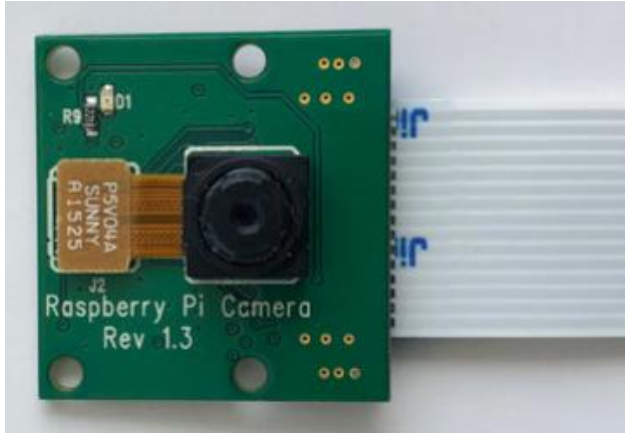
https://www.gps-coordinates.net/

Global Navigation Satellite Systems

- The cheap GPS receivers still have an error of several meters, especially near buildings that affects positioning accuracy.
- **Differential GPS (DGPS)** is one method to improve accuracy by adding a stationary reference station with a second GPS receiver that is linked to the GPS receiver on the mobile robot through a wireless connection.
- As both receivers of a DGPS suffer from the same positioning errors (pseudoranges), the stationary receiver can send a correction signal to the mobile GPS receiver to improve its positioning to several centimeters.

Digital Image Sensors

- Digital cameras are the most complex sensors used in robotics
- For mobile robot applications, we are interested in a **high frame rate**, because our robot is moving, and we want updated sensor data as fast as possible.
- Since there is always a trade-off between high frame rate and high resolution, we are not so much concerned with camera resolution.
- The image resolution must be **high enough** to detect a desired object from a specified distance.
- When the object in the distance is reduced to a mere few pixels, then this is not sufficient for a detection algorithm.



Raspberry Pi camera module