The Embedded Computing Platform

Input/Output Interfaces

José Costa

Software for Embedded Systems

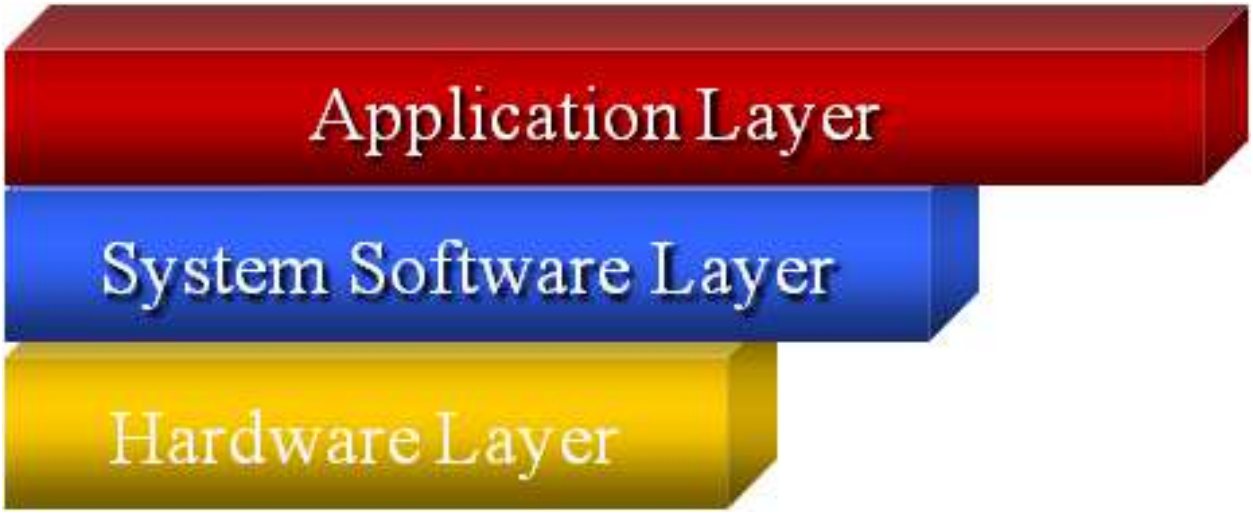
Departamento de Engenharia Informática (DEI)
Instituto Superior Técnico

2015-09-28

Outline

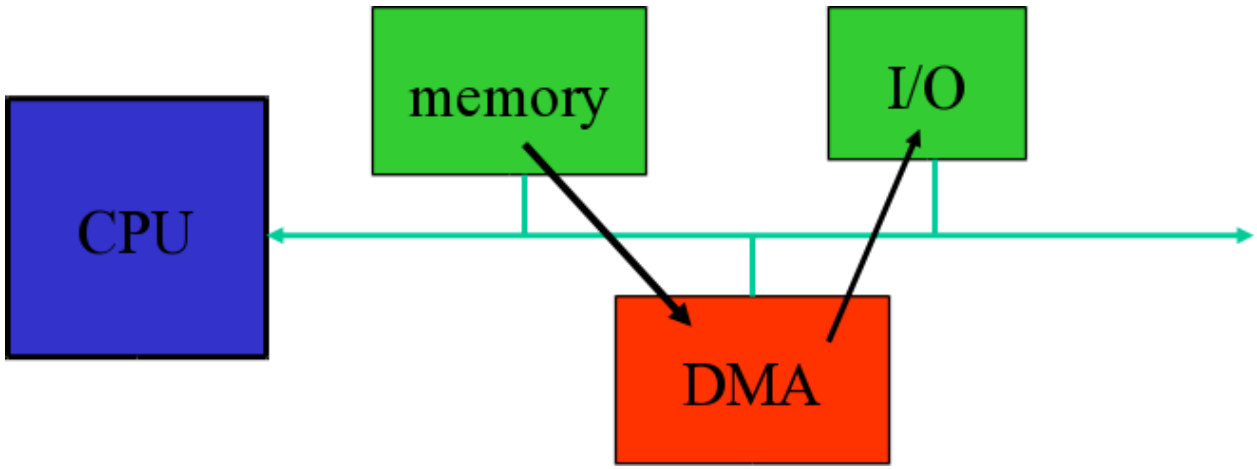
- IO Interfaces
  - Timers and Counters
  - Watchdog Timer
  - Keyboards
  - LEDs, LCDs
  - ADC and DACs
  - Temperature Sensor
  - Sound Sensors
  - IR Sensor, Photoelectric Sensors
  - GPS, Compass, Accelerometer, Gyroscope
  - Ambient Light Sensors

Embedded Systems Model



Input/Output (I/O) Interfaces

- I/O is a fundamental function of the embedded systems platform
- I/O encapsulates the interaction with the real (physical) world
- I/O servicing is done by the cooperation of hardware + software



Basic I/O Capabilities

- Time measurement
- Event synchronization
- Analog interfacing
- Digital I/O
- Sensors

I/O Devices

- serial links
- timers and counters
- keyboards
- displays
- analog I/O

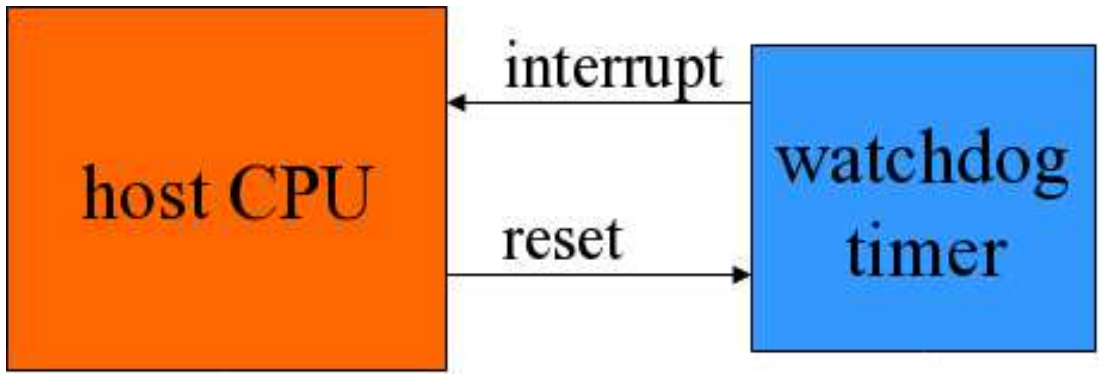
- Found as on-chip devices in micro-controllers
- or implemented separately

Timers and Counters

- Very similar
  - a timer is incremented/decremented by a periodic signal
  - a counter is incremented by an asynchronous, occasional signal
- Very often called counter/timer
- Rollover causes interrupt
- Has reset register to reset the timer
- Most provide cyclic and acyclic modes

Watchdog Timer

- Watchdog timer is periodically reset by system timer
- If watchdog is not reset
  - usually due to software flaw or hardware problem
  - it generates an interrupt to reset the host
    - rather than diagnose problem
    - thus, system is put in operation as quickly as possible



Example of Watchdog (1/6)

```
 1: main() {
 2:   while (1) {
 3:     TaskA();
 4:     if (B) {
 5:       TaskB();
 6:       if (C) {
 7:         TaskC();
 8:       }
 9:     } else {
10:       TaskD();
11:       if (E) {
12:         TaskE();
13:       }
14:     }
15:   }
16: }
```

An embedded system with a watchdog sentinel has a program with the following duration of tasks:

Task	Duration
A	10ms
B	60ms
C	30ms
D	30ms
E	40ms

Knowing that the timeout of the watchdog timer is 75ms, which of the alternatives below is the best option of where to insert the function call to reset the watchdog?

A:	B:	C:	D:
After lines 2, 5	After lines 3, 6, 12	After lines 2, 3, 6	After lines 3, 5, 7, 11

Example of Watchdog (2/6)

```
 1: main() {
 2:   while (1) {
        WatchDog();
 3:     TaskA();
 4:     if (B) {
 5:       TaskB();
          WatchDog();
 6:       if (C) {
 7:         TaskC();
 8:       }
 9:     } else {
10:       TaskD();
11:       if (E) {
12:         TaskE();
13:       }
14:     }
15:   }
16: }
```

An embedded system with a watchdog sentinel has a program with the following duration of tasks:

Task	Duration
A	10ms
B	60ms
C	30ms
D	30ms
E	40ms

Knowing that the timeout of the watchdog timer is 75ms, which of the alternatives below is the best option of where to insert the function call to reset the watchdog?

A:	B:	C:	D:
After lines 2, 5	After lines 3, 6, 12	After lines 2, 3, 6	After lines 3, 5, 7, 11

Example of Watchdog (3/6)

```
 1: main() {
 2:   while (1) {
 3:     TaskA();
        WatchDog();
 4:     if (B) {
 5:       TaskB();
 6:       if (C) {
            WatchDog();
 7:         TaskC();
 8:       }
 9:     } else {
10:       TaskD();
11:       if (E) {
12:         TaskE();
            WatchDog();
13:       }
14:     }
15:   }
16: }
```

An embedded system with a watchdog sentinel has a program with the following duration of tasks:

Task	Duration
A	10ms
B	60ms
C	30ms
D	30ms
E	40ms

Knowing that the timeout of the watchdog timer is 75ms, which of the alternatives below is the best option of where to insert the function call to reset the watchdog?

A:	B:	C:	D:
After lines 2, 5	After lines 3, 6, 12	After lines 2, 3, 6	After lines 3, 5, 7, 11

Example of Watchdog (4/6)

```
 1: main() {
 2:   while (1) {
        WatchDog();
 3:     TaskA();
        WatchDog();
 4:     if (B) {
 5:       TaskB();
 6:       if (C) {
            WatchDog();
 7:         TaskC();
 8:       }
 9:     } else {
10:       TaskD();
11:       if (E) {
12:         TaskE();
13:       }
14:     }
15:   }
16: }
```

An embedded system with a watchdog sentinel has a program with the following duration of tasks:

Task	Duration
A	10ms
B	60ms
C	30ms
D	30ms
E	40ms

Knowing that the timeout of the watchdog timer is 75ms, which of the alternatives below is the best option of where to insert the function call to reset the watchdog?

A:	B:	C:	D:
After lines 2, 5	After lines 3, 6, 12	After lines 2, 3, 6	After lines 3, 5, 7, 11

Example of Watchdog (5/6)

```
 1: main() {
 2:   while (1) {
 3:     TaskA();
        WatchDog();
 4:     if (B) {
 5:       TaskB();
          WatchDog();
 6:       if (C) {
 7:         TaskC();
            WatchDog();
 8:       }
 9:     } else {
10:       TaskD();
11:       if (E) {
            WatchDog();
12:         TaskE();
13:       }
14:     }
15:   }
16: }
```

An embedded system with a watchdog sentinel has a program with the following duration of tasks:

Task	Duration
A	10ms
B	60ms
C	30ms
D	30ms
E	40ms

Knowing that the timeout of the watchdog timer is 75ms, which of the alternatives below is the best option of where to insert the function call to reset the watchdog?

A:	B:	C:	D:
After lines 2, 5	After lines 3, 6, 12	After lines 2, 3, 6	After lines 3, 5, 7, 11

Example of Watchdog (6/6)

TÉCNICO LISBOA

Option B

```
main() {
  while (1) {
    TaskA();
    WatchDog();
    if (B) {
      TaskB();
      if (C) {
        WatchDog();
        TaskC();
      }
    } else {
      TaskD();
      if (E) {
        TaskE();
        WatchDog();
      }
    }
  }
}
```

Option C

```
main() {
  while (1) {
    WatchDog();
    TaskA();
    WatchDog();
    if (B) {
      TaskB();
      if (C) {
        WatchDog();
        TaskC();
      }
    } else {
      TaskD();
      if (E) {
        TaskE();
      }
    }
  }
}
```

Option D

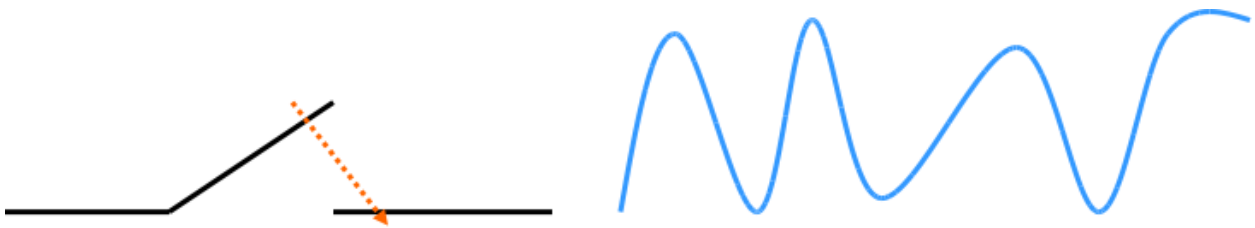
```
main() {
  while (1) {
    TaskA();
    WatchDog();
    if (B) {
      TaskB();
      WatchDog();
      if (C) {
        TaskC();
        WatchDog();
      }
    } else {
      TaskD();
      if (E) {
        WatchDog();
        TaskE();
      }
    }
  }
}
```

Keyboards

- Basically an array of switches
- Could include some internal logic
  - to simplify interface with microprocessor
- Uses mechanical contact to make or break an electric circuit
- Bouncing is major problem

Switch Debouncing
Interfacing With the Physical World

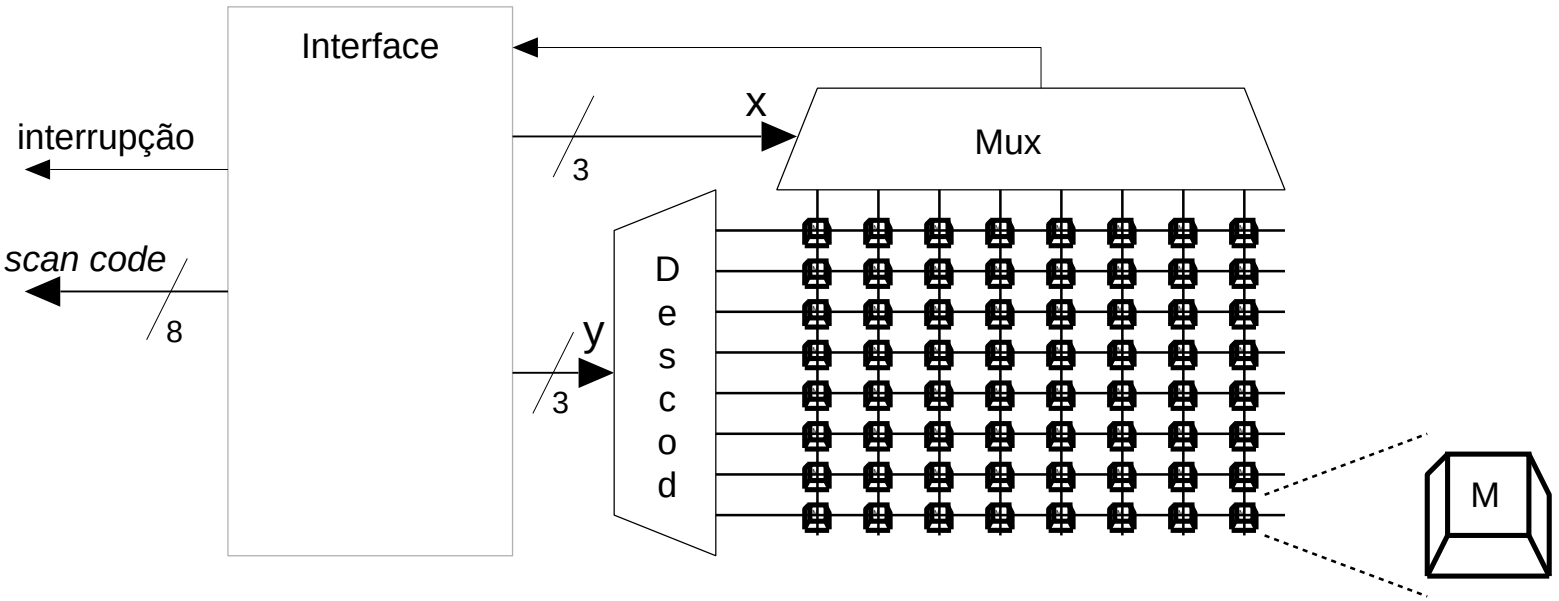
- A switch must be debounced to multiple contacts caused by eliminate mechanical bouncing:



- Hardware debouncing circuit can be built using one-shot timer
- Software can also be used

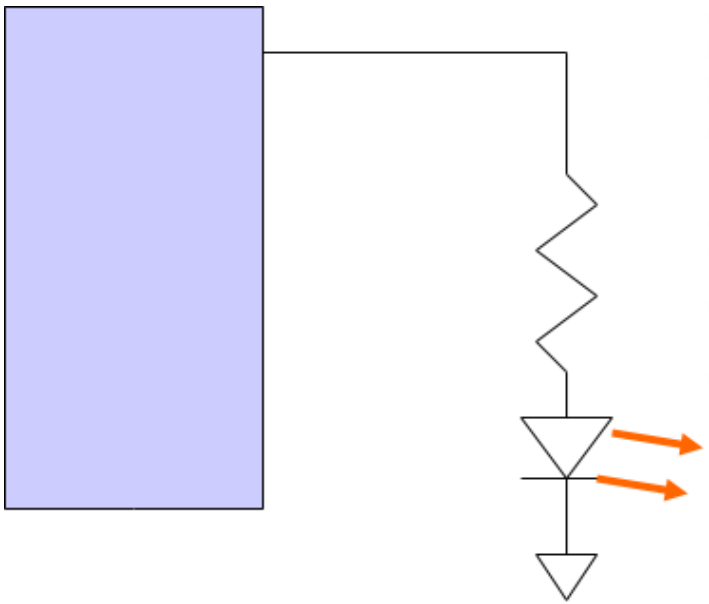
Keyboards

- The typical PC keyboard
- An array of switches is read by an encoder
- N-key rollover remembers multiple key depressions



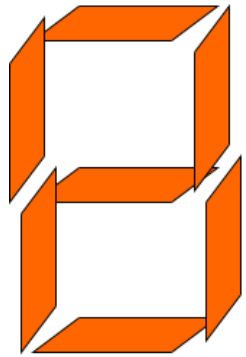
LEDs

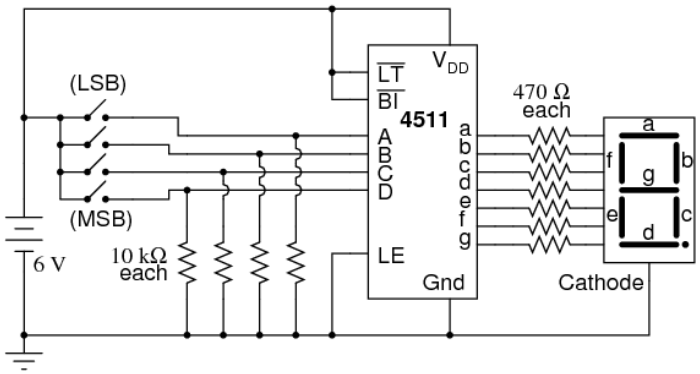
- Often used as simple displays by themselves
- Arrays of LEDs may form more complex displays
- Must use resistor to limit current:

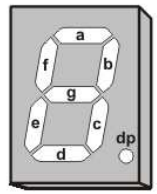


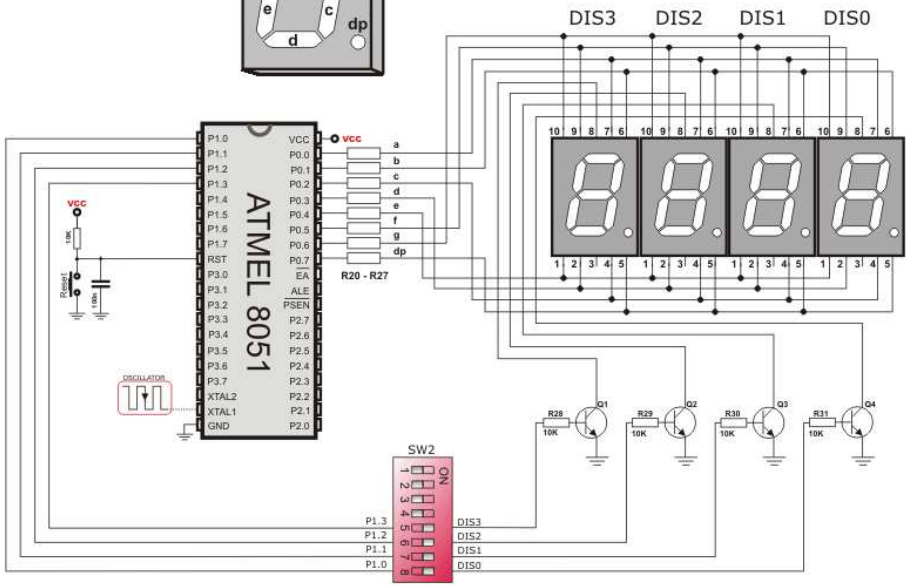
7-Segment LED Display

- May use parallel or multiplexed input.







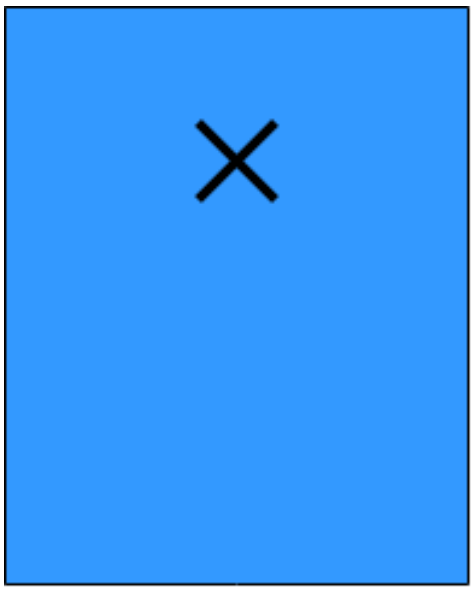


Types of High-Resolution Display

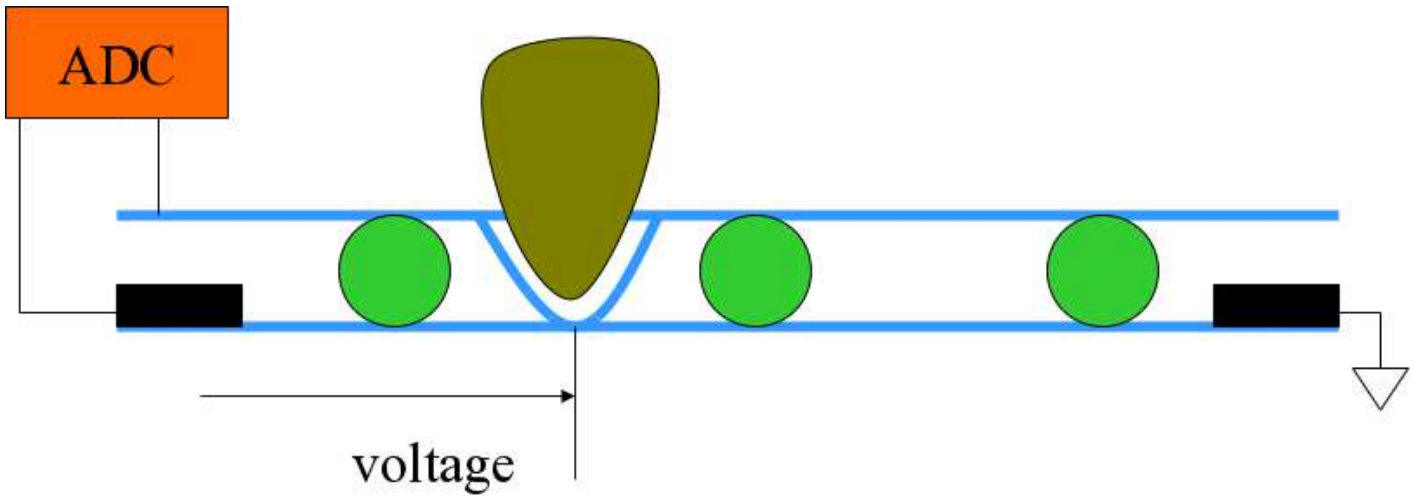
- Liquid crystal display (LCD) is dominant form
  - modern LCDs use active matrix
- Plasma, OLED, etc.
- Frame buffer holds current display contents
  - Written by processor
  - Read by video

Touchscreen

- Includes input and output device
  - output device below
  - top conductive sheet is flexible
- Input device is a two-dimensional voltmeter:



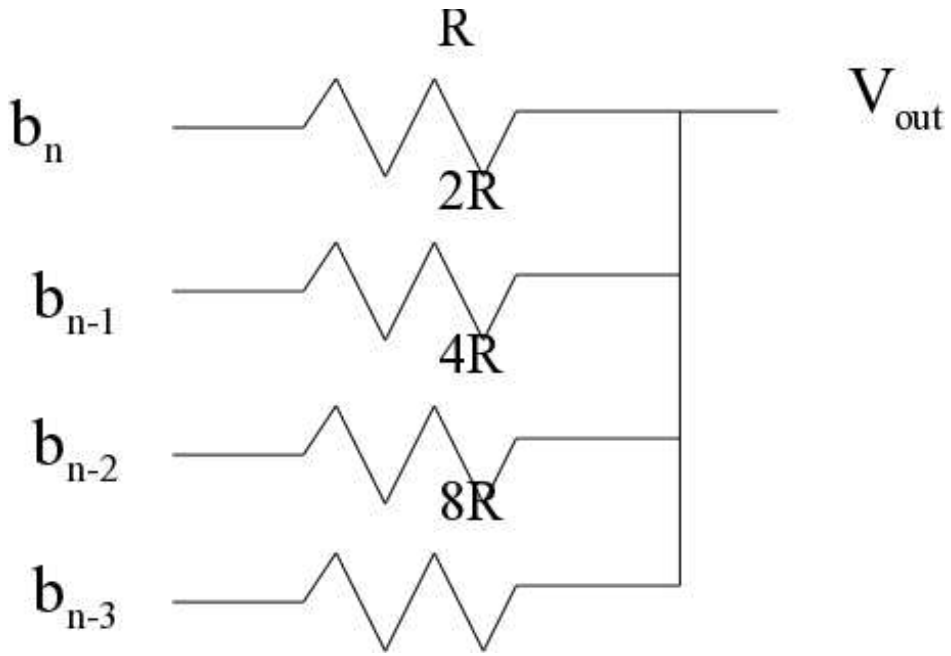
Touchscreen Position Sensing



- Voltage applied across the sheet
- An ADC is used to measure voltage and thus obtain the position
- Voltage alternates between x and y positions

Digital-to-Analog Conversion

- Use resistor tree:

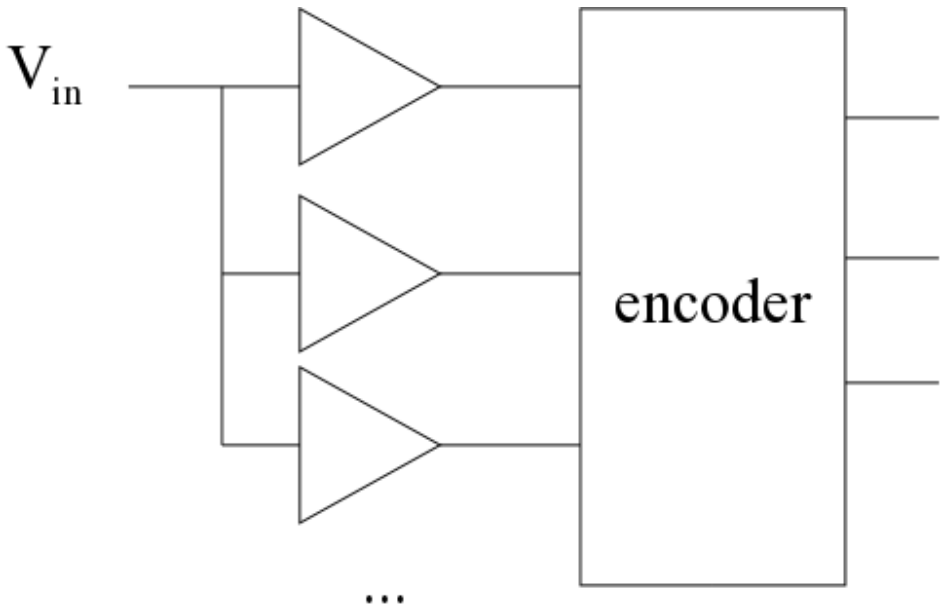


Analog-to-Digital Conversion

- A simple but good example of hardware/software balance
- Hardware only
  - Flash conversion
- Hardware + software (program or dedicated controller)
  - Successive approximation
- Mostly software
  - Dual-slope conversion

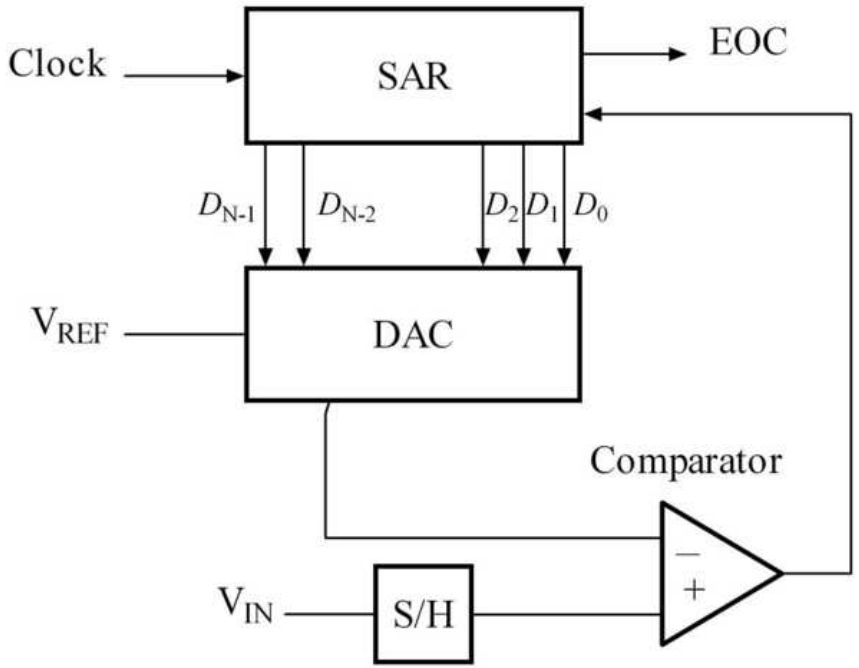
Flash A/D Conversion

- Very fast
- N-bit result requires $2^n - 1$ comparators
  - Expensive circuit



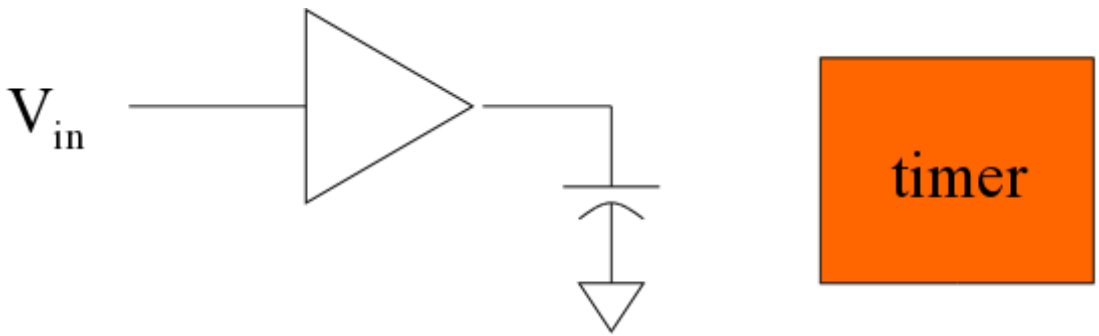
Successive Approximation

- Uses a DAC
- Compares the output of the DAC with sampled value
- One bit is decided in each step



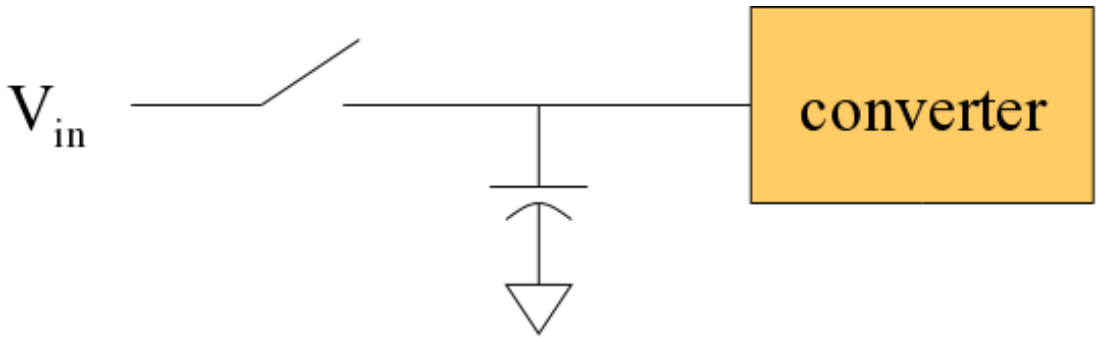
Dual-Slope Conversion

- Use counter to time required to charge/discharge capacitor
- Charging, then discharging eliminates non-linearities
- Not very fast
  - can be made faster by using less resolution



Sample-and-Hold

- Required in any A/D:



ADC Using a D/A Converter and An Analog Comparator (1/4)

```
#define MAX 256
int i;
for (i = 0; i < MAX; i = i+1) {
  output_to_DAC (i);
  wait (DA_conversion_time);
  if (input_from_comparator () >= 0)
    break;
} /* i holds the result */
```

Interrupt service routine is triggered by a periodic interrupt signal.
D/A converter contiguous step: $\text{Data}_{\text{In}} \rightarrow V_{\text{Out}}$: 200 ns
D/A converter maximum step: $\text{Data}_{\text{In}} \rightarrow V_{\text{Out}}$: 1 µ s
Comparator: $V_{\text{in}} \rightarrow V_{\text{out}}$: 100 ns

---

Value of DA_conversion_time?

- output_to_DAC + comparator: 200ns + 100ns = 300ns

ADC Using a D/A Converter and An Analog Comparator (2/4)

```
#define MAX 256
int i;
for (i = 0; i < MAX; i = i+1) {
  output_to_DAC (i);
  wait (DA_conversion_time);
  if (input_from_comparator () >= 0)
    break;
} /* i holds the result */
```

Interrupt service routine is triggered by a periodic interrupt signal.
D/A converter contiguous step: $\text{Data}_{\text{In}} \rightarrow V_{\text{Out}}$: 200 ns
D/A converter maximum step: $\text{Data}_{\text{In}} \rightarrow V_{\text{Out}}$: 1 µ s
Comparator: $V_{\text{in}} \rightarrow V_{\text{out}}$: 100 ns

---

DA_conversion_time = 2 µ s. Maximum frequency of the conversion interrupt?

- 256 iterations x ...
- 256 x 2 µ s = 1.95 kHz

ADC Using a D/A Converter and An Analog Comparator (3/4)

```
#define MAX 256
int i;
for (i = 0; i < MAX; i = i+1) {
  output_to_DAC (i);
  wait (DA_conversion_time);
  if (input_from_comparator () >= 0)
    break;
} /* i holds the result */
```

Interrupt service routine is triggered by a periodic interrupt signal.
D/A converter contiguous step: $\text{Data}_{\text{In}} \rightarrow V_{\text{Out}}$: 200 ns
D/A converter maximum step: $\text{Data}_{\text{In}} \rightarrow V_{\text{Out}}$: 1 µ s
Comparator: $V_{\text{in}} \rightarrow V_{\text{out}}$: 100 ns

---

The signal to be converted has an average value of 30% of the conversion range [0,Vmax]. Considering again DA_conversion_time = 2 µ s , evaluate the average execution time of the interrupt routine.

- 30% x ...
- 30% x 256 * 2 µ s = 154 µ s

ADC Using a D/A Converter and An Analog Comparator (4/4)

How to optimize conversion time?

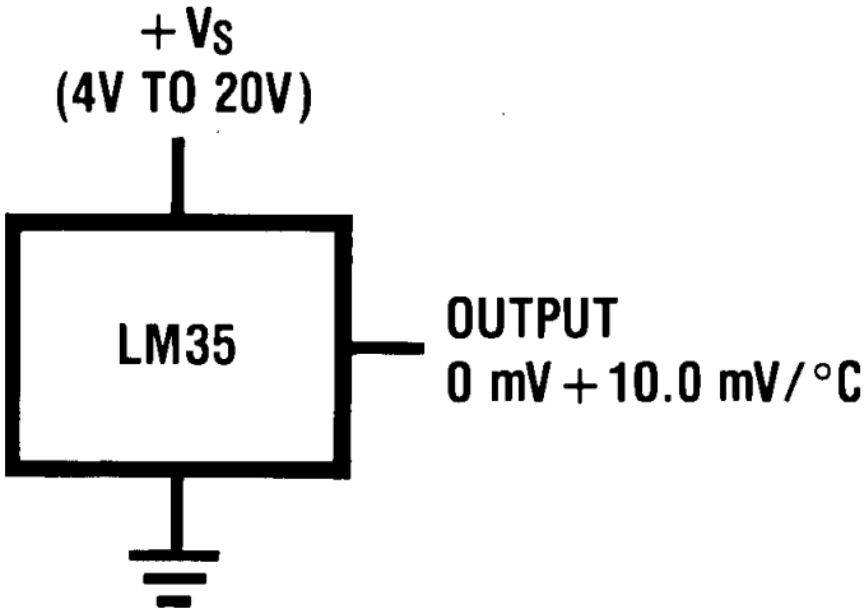
- Use faster hardware
- Start with i = MAX/2
- Use previous value of i
- Use powers of 2 - Best!

Sensors

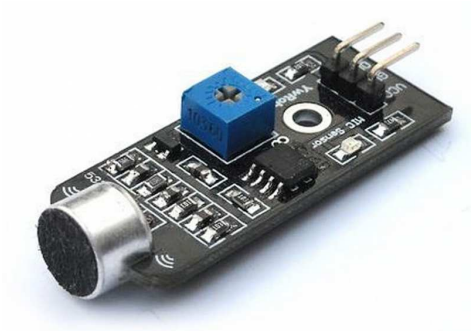
- Sensors play an important role in modern electronic devices and are used in many fields
- They are intended to convert physical quantities into signals
- Give the possibility to the devices to sense and gather information about the environment that surrounds them
- Are present in everyday objects
- The number of sensor types is enormous and each sensor type is intended to detect a specific physical quantity

Temperature Sensor

- Output voltage gives temperature
- Must use an ADC



Sound Sensors



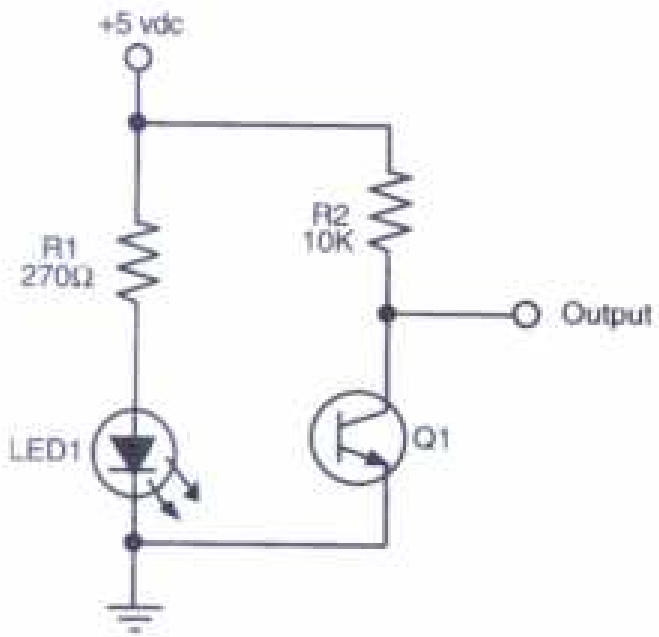
- Analog devices that convert acoustical waves into electrical signals
- Composed by a microphone and an amplifier
  - The microphone detects noise and produces signals with values proportional to the noises detected
  - Signals are then amplified by the amplifier since the microphone signals are to small to be handled by the controller pins
- Often used in security devices, for voice recognition, and in robotics as an interaction interface

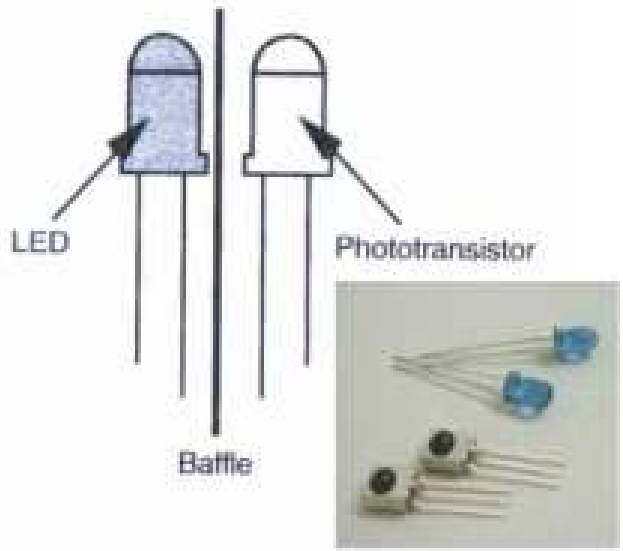
Motion & Location Sensors

- Sensors used to detect the presence or absence of objects
- Do not need any physical contact with objects
- Wide range of sensor types:
  - Inductive
  - Capacitive
  - Photoelectric
  - Magnetic
  - Sonar
  - ...

IR Sensor

- Emitter emits infrared (IR) rays
- IR rays are detected by a detector
- Can be used as a remote controller or proximity sensor

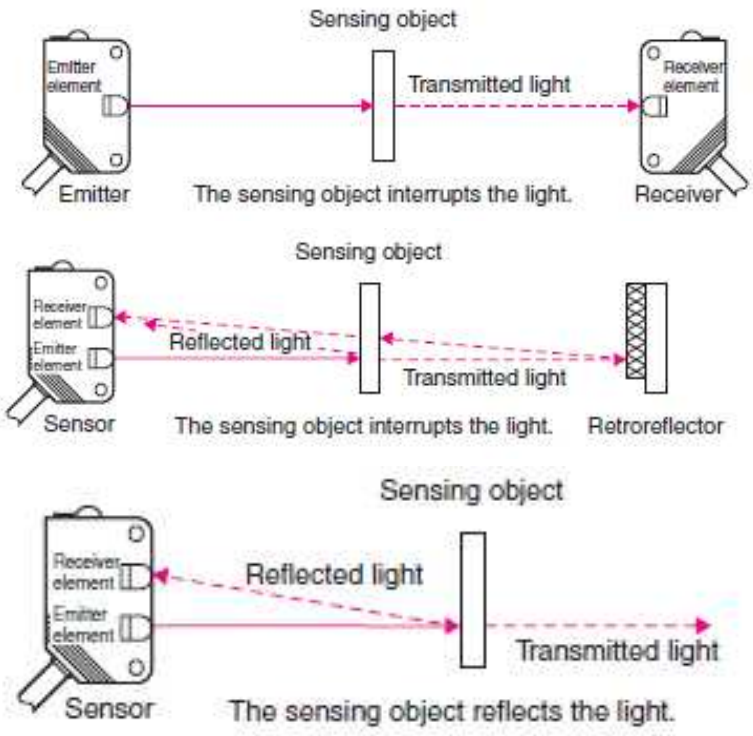




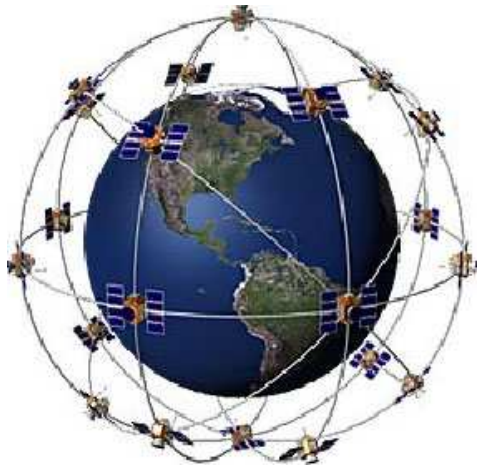
Photoelectric Sensors

- Photoelectric sensors are used to detect the distance, absence, or presence of an object;
- Are composed by a light transmitter and a light sensor;

- There are three types:
  - Through-Beam sensors
  - Retro-Reflective sensors
  - Diffuse-reflective sensors



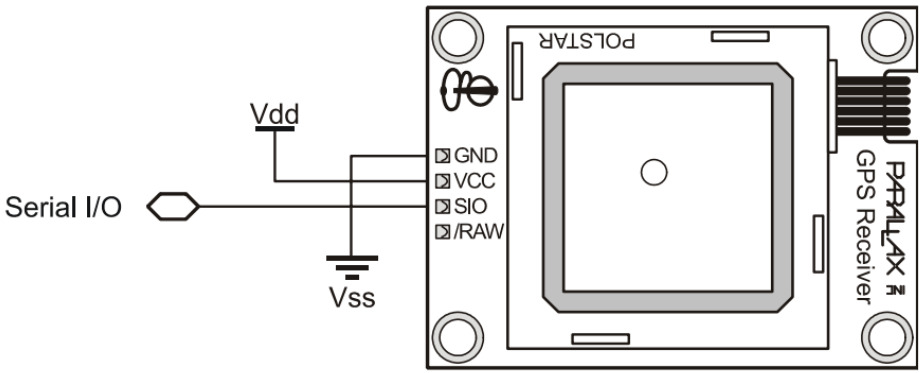
GPS (Global Positioning System)



- Radio navigation system that allows land, sea, and airborne users to determine their exact location, velocity, and time 24 hours a day, in all weather conditions, anywhere in the world
  - 24 GPS satellites are in orbit at 10,600 miles above the earth
    - The satellites are spaced so that from any point on earth, four satellites will be above the horizon
    - Each satellite contains a computer, an atomic clock, and a radio
  - GPS receiver contains a computer that "triangulates" its own position by getting bearings from three of the four satellites
    - result is provided in the form of a geographic position (longitude and latitude) within a few meters

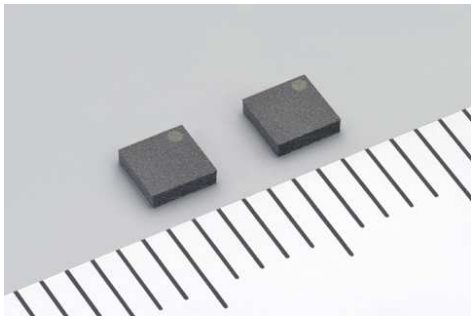
GPS Sensor

- Serial communication used to send commands to the GPS and to receive data
- Two modes of operation
  - smart mode
  - raw mode



Compass

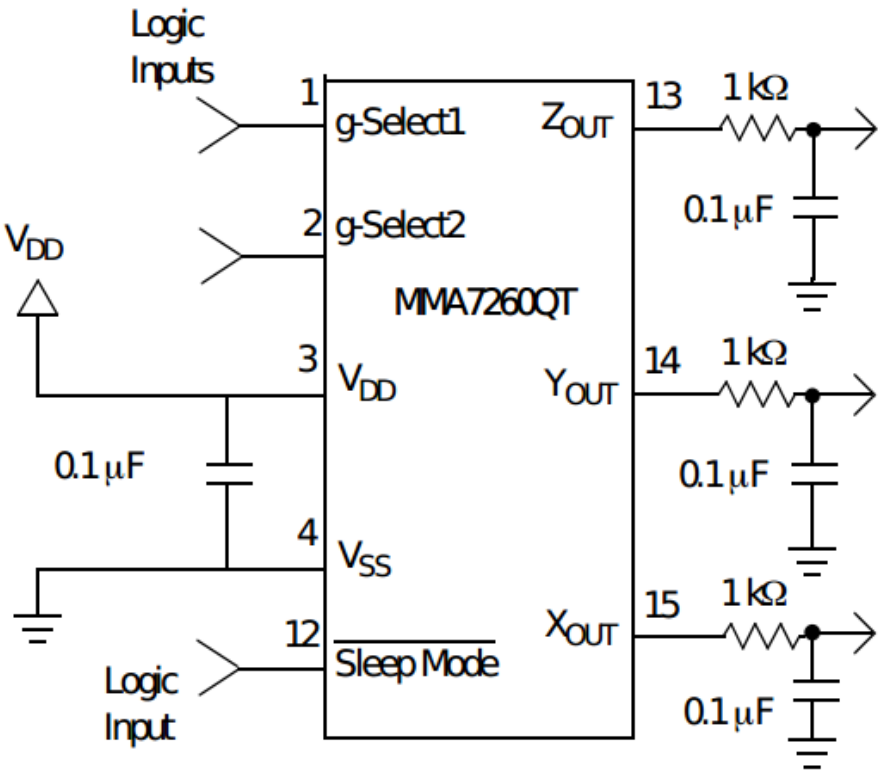
- The magnetic field parallel to the earth surface points towards the magnetic north pole. Because of this the compass needs to be held parallel to the surface



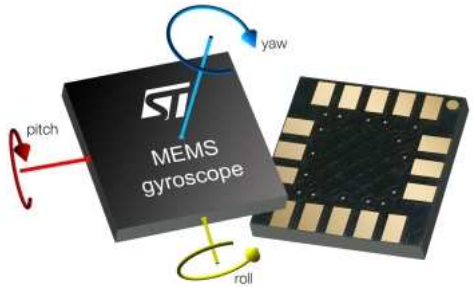
- Electronic compass sensors were invented because common compass would make the electronic devices capabilities useless
- Requirements:
  - They have to measure two orthogonal axes X and Y of the earth magnetic field in order to detect the parallelism to the earths surface
  - To be perming, insensitive to the thermal and magnetic shock
  - Have low power consumption

Accelerometer

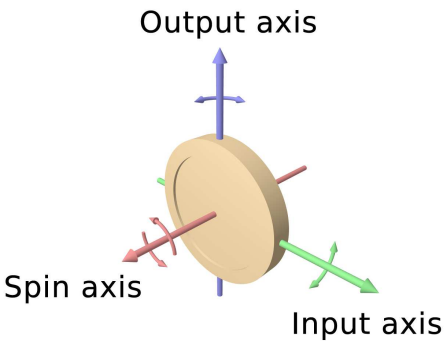
- Logic inputs define range and sensitivity
- Outputs give acceleration in the 3 axis



Gyroscope



- Gyroscopes are used in smart devices to sense their orientation and rotation
- Allows the device to sense motion on six axes: left, right, up, down, forward and backward, as well as roll, pitch and yaw rotations



- Types:
  - Micro Electro-Mechanical System (MEMS)
  - Fiber optic gyroscope (FOG)
  - Vibrating structure gyroscope (VSG)
  - Dynamically tuned gyroscope (DTG)
  - Ring Laser gyroscope

Gyroscope

Vibrating structure gyroscope

Micro Electro-Mechanical System

Ring laser gyroscope

Ambient Light Sensors

- Sensors that convert light into either current or voltage
  - Photoelectric cells
  - Photo diodes
  - Photo transistors
  - Photo ICs

- Ambient Light sensors are largely related to the use of Liquid Crystal Displays (LCD) and its associated backlight systems
  - These sensors can be used to optimize the operation of the backlight dimming more or less, depending on the conditions, and thus saving the power consumption

Other interfaces

- Communication
  - Serial
  - USB
  - Ethernet
  - Wifi
- Cards
  - SD cards
  - Compact Flash
- ...

Outline

- IO Interfaces
  - Timers and Counters
  - Watchdog Timer
  - Keyboards
  - LEDs, LCDs
  - ADC and DACs
  - Temperature Sensor
  - Sound Sensors
  - IR Sensor, Photoelectric Sensors
  - GPS, Compass, Accelerometer, Gyroscope
  - Ambient Light Sensors

References

- Computers as Components: Principles of Embedded Computing System Design , Marilyn Wolf. Morgan Kaufman. Chs. 4.3

Next class

Input/Output Interfaces and Service