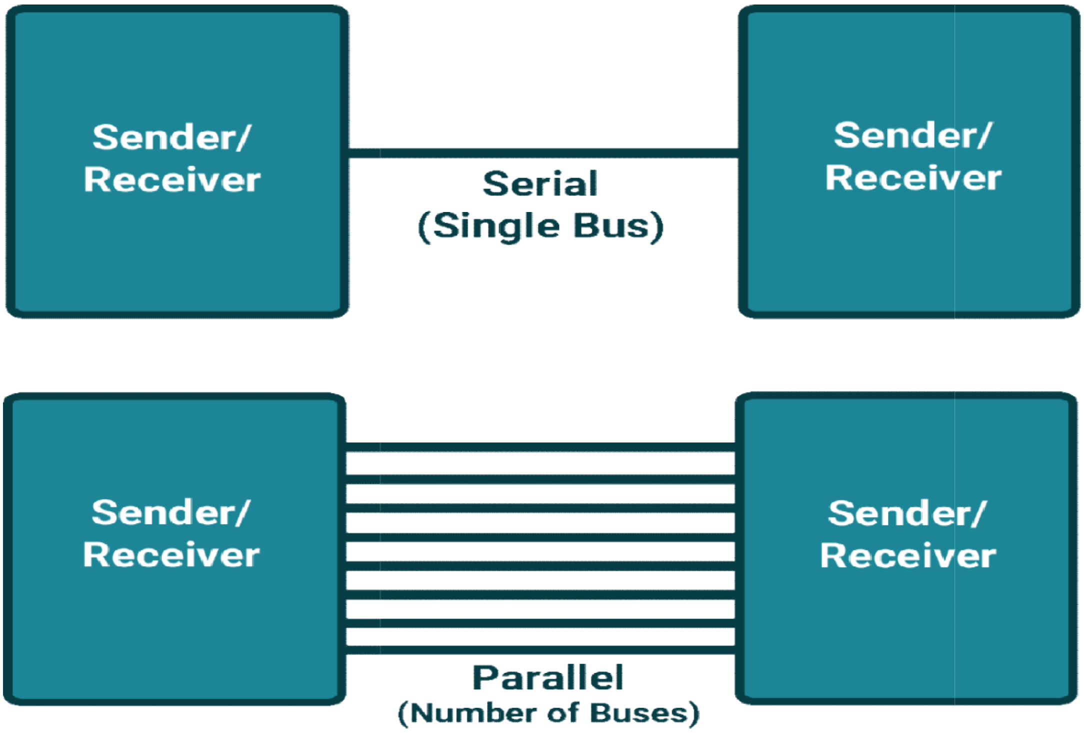
Module-2-UART – Sensors – Actuators

UART - Introduction



8051 UART Serial vs Parallel Communication

Serial communication means to transfer data bit by bit serially at a time, whereas in parallel communication, the number of bits that can be transferred at a time depends upon the number of data lines available for communication.

Two methods of serial communication are

- Synchronous Communication: Transfer of bulk data in the framed structure at a time
- Asynchronous Communication: Transfer of a byte data in the framed structure at a time

8051 has built-in UART with RXD (serial data receive pin) and TXD (serial data transmit pin) on PORT3.0 and PORT3.1 respectively.

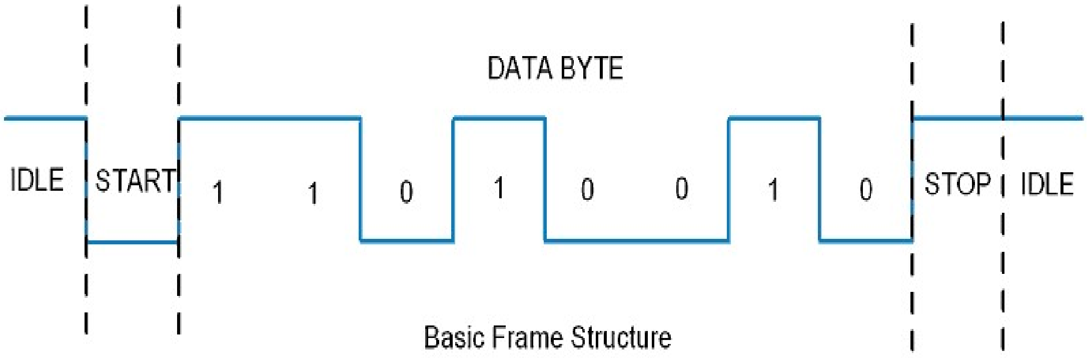
Asynchronous communication

Asynchronous serial communication is widely used for byte-oriented transmission.

Frame structure in Asynchronous communication:

- **START bit:** It is a bit with which serial communication starts and it is always low.
- **Data bits packet**: Data bits can be 5 to 9 bits packet. Normally we use 8 data bit packet, which is always sent after the START bit.
- **STOP bit**: This is one or two bits. It is sent after the data bits packet to indicate the end of the frame. The stop bit is always logic high.

In an asynchronous serial communication frame, the first START bit followed by the data byte and at last STOP bit forms a 10-bit frame. Sometimes the last bit is also used as a parity bit.



8051

Serial Frame Structure

Data transmission rate

The data transmission rate is measured in bits per second (bps). In the binary system, it is also called a baud rate (number of signal changes per second). Standard baud rates supported are 1200, 2400, 4800, 19200, 38400, 57600, and 115200. Normally most of the time 9600 bps is used when speed is not a big issue.

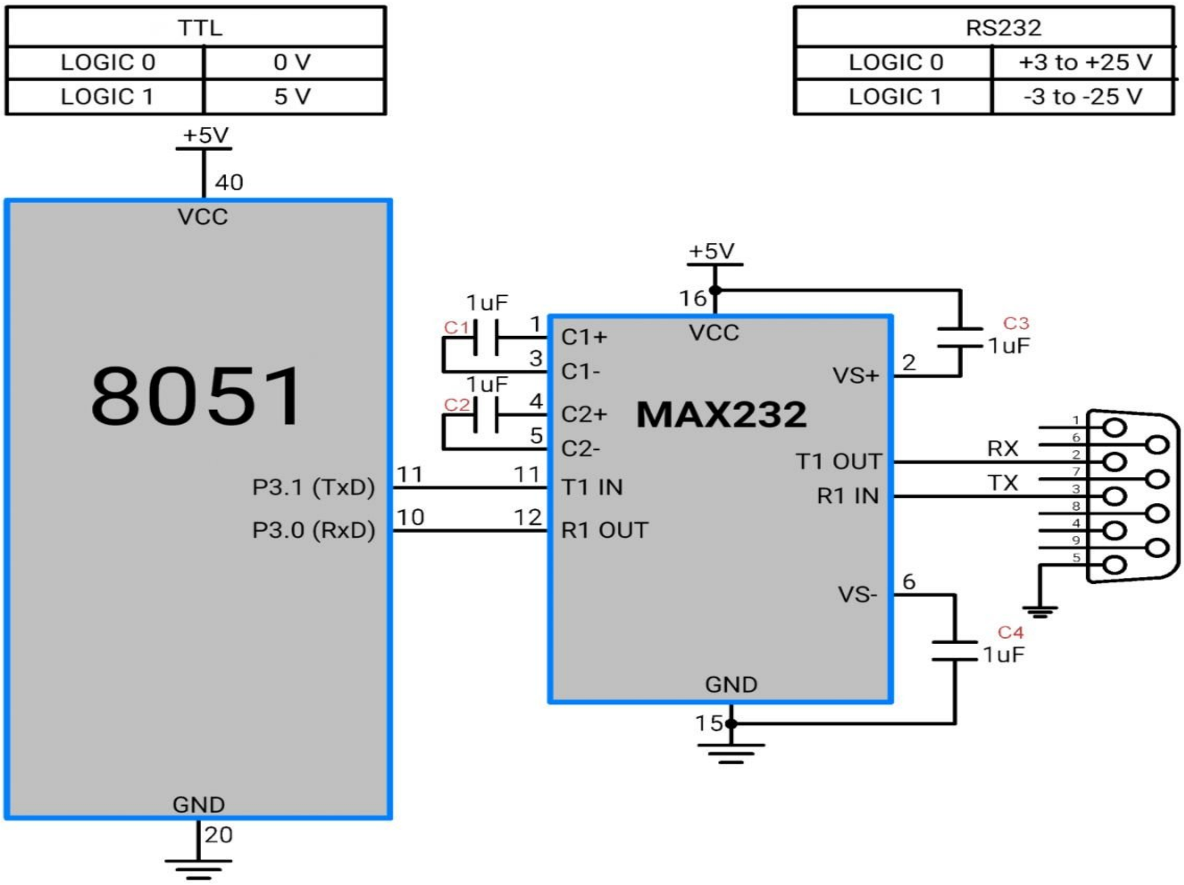
Interface standard

- ✓ 8051 serial communication has TTL voltage level which are 0 v for logic 0 and 5 v for logic 1.

  - ✓ In computers and most of the old devices for serial communication, RS232 protocol with DB9 connector is used.
  - ✓ RS232 serial communication has different voltage levels than 8051 serial communication. i.e. +3 v to +25 v for logic zero and -3 v to -25 v for logic 1.

- ✓ So to communicate with RS232 protocol, we need to use a voltage level converter like MAX232 IC.
- ✓ Although there are 9 pins in the DB9 connector, we don't need to use all the pins. Only 2nd Tx (Transmit), 3rd Rx(Receive), and 5th GND pin need to be connected.

8051 Serial Interface Diagram



TTL	
LOGIC 0	0 V
LOGIC 1	5 V

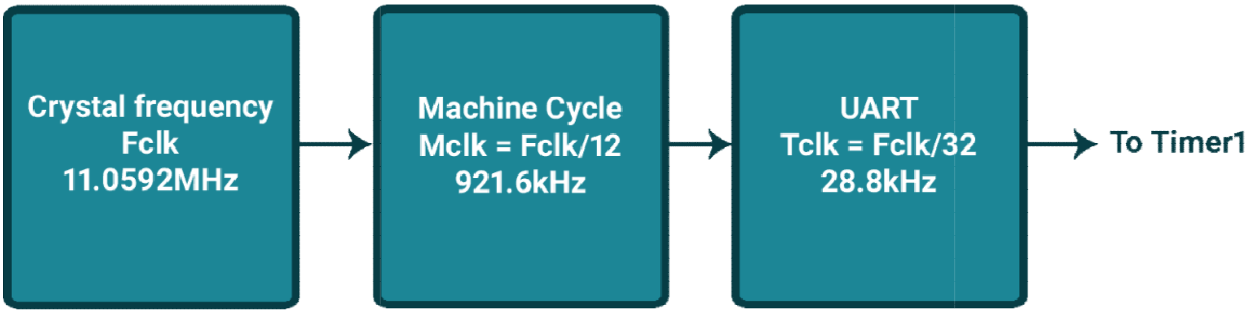
RS232	
LOGIC 0	+3 to +25 V
LOGIC 1	-3 to -25 V

MAX232 interfacing With 8051 MCU

8051 UART Programming

Baud Rate calculation:

- ✓ To meet the standard baud rates generally crystal with 11.0592 MHz is used.
- ✓ As we know, 8051 divides crystal frequency by 12 to get a machine cycle frequency of 921.6 kHz.
- ✓ The internal UART block of 8051 divides this machine cycle frequency by 32, which gives the frequency of 28800 Hz which is used by UART.
- ✓ To achieve a baud rate of 9600, again 28800 Hz frequency should be divided by 3.
- ✓ This is achieved by using Timer1 in mode-2 (auto-reload mode) by putting 253 in TH1 (8-bit reg.)
- ✓ So 28800 Hz will get divided by 3 as the timer will overflow after every 3 cycles**.**
- ✓ we can achieve different baud rates by putting the division factor in the TH1 register.



8051 Microcontroller Baud Rate calculation

Division factor to achieve different baud rates

Baud Rate	TH1 (Hex)
9600	FD
4800	FA
2400	F4
1200	E8

8051 Serial communication Registers

SBUF: Serial Buffer Register

This is the serial communication data register used to transmit or receive data through it.

7	6	5	4	3	2	1	0	
								SBUF

SCON: Serial Control Register

Serial control register SCON is used to set serial communication operation modes. Also it is used to control transmit and receive operations.

7	6	5	4	3	2	1	0	
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON

Bit 7:6 - SM0:SM1: Serial Mode Specifier

Mode	SM0	SM1	Mode
0	0	0	1/12 of Osc frequency shift register mode fixed baud rate
1	0	1	8-bit UART with timer 1 determined baud rate
2	1	0	9-bit UART with 1/32 of Osc fixed baud rate
3	1	1	9-bit UART with timer 1 determined baud rate

Normally mode-1 (SM0 =0, SM1=1) is used with 8 data bits, 1 start bit, and 1 stop bit.

Bit 5 - SM2: for Multiprocessor Communication. This bit enables a multiprocessor communication feature in mode 2 & 3.

Bit 4 - REN: Receive Enable

1 = Receive enable
0 = Receive disable

Bit 3 - TB8: 9th Transmit Bit

This is the 9th bit which is to be transmitted in mode 2 & 3 (9-bit mode)

Bit 2 - RB8: 9th Receive Bit
This is the 9th received bit in mode 2 & 3 (9-bit mode), whereas in mode 1 if SM2 = 0 then RB8 hold the stop bit that received

Bit 1 - TI: Transmit Interrupt Flag
This bit indicates the transmission is complete and gets set after transmitting the byte from the buffer. Normally TI (Transmit Interrupt Flag) is set by hardware at the end of the 8th bit in mode 0 and at the beginning of stop bit in other modes.

Bit 0 – RI: Receive Interrupt Flag
This bit indicates reception is complete and gets set after receiving the complete byte in the buffer. Normally RI (Receive Interrupt Flag) is set by hardware in receiving mode at the end of the 8th bit in mode 0.

8051 Microcontroller Programming steps

1. Configure Timer 1 in auto-reload mode.
2. Load TH1 with value as per required baud rate e.g. for 9600 baud rate load 0xFD. (-3 in decimal).
3. Load SCON with serial mode and control bits. e.g. for mode 1 and enable reception, load 0x50.
4. Start timer1 by setting TR1 bit to 1.
5. Load transmitting data in the SBUF register.
6. Wait until loaded data is completely transmitted by polling the TI flag.
7. When the TI flag is set, clear it, and repeat from step 5 to transmit more data.

Example

Let's Program 8051 (here AT89C51) to send character data “test” serially at 9600 baud rate in mode 1

8051 Serial Program for serial data transmit

```
#include <reg51.h>            /* Include x51 header file */
void UART_Init()
{
        TMOD = 0x20;          /* Timer 1, 8-bit auto reload mode */
        TH1 = 0xFD;           /* Load value for 9600 baud rate */
        SCON = 0x50;          /* Mode 1, reception enable */
        TR1 = 1;               /* Start timer 1 */
}

void Transmit_data(char tx_data)
{
        SBUF = tx_data;               /* Load char in SBUF register */
        while (TI==0);        /* Wait until stop bit transmit */
        TI = 0;               /* Clear TI flag */
}

void String(char *str)
{
        int i;
        for(i=0;str[i]!=0;i++)        /* Send each char of string till the
NULL */
        {
                Transmit_data(str[i]); /* Call transmit data function */
        }
}
void main()
{
        UART_Init();          /* UART initialize function */
        String("test");       /* Transmit 'test' */
        while(1);
}
```

Sensors and Actuators

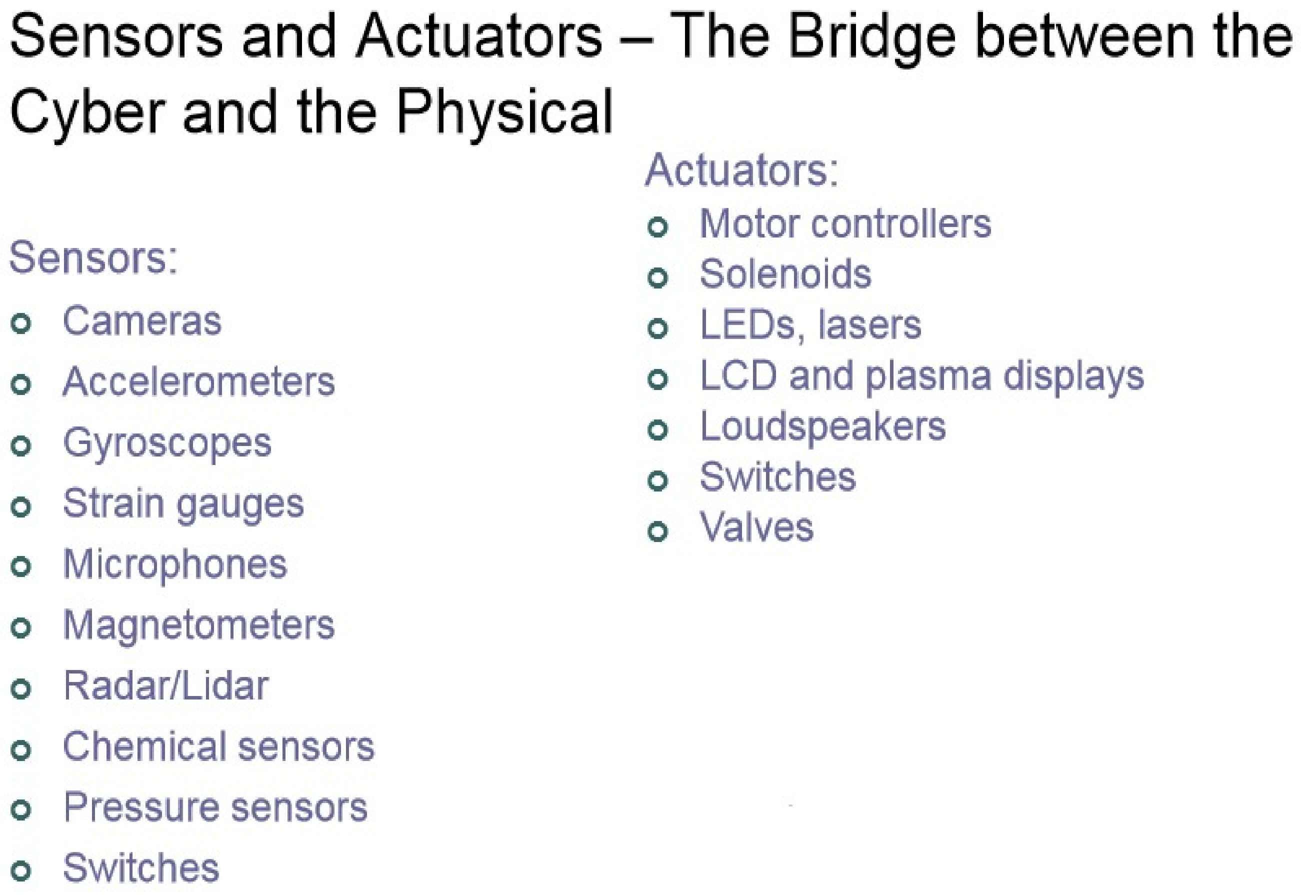
What is a sensor?

A sensor is a device that measures a physical quantity

- Input / “Read from physical world”

What is an actuator?

- An actuator is a device that modifies a physical quantity Output / “Write to physical world”



Sensors and Actuators – The Bridge between the Cyber and the Physical

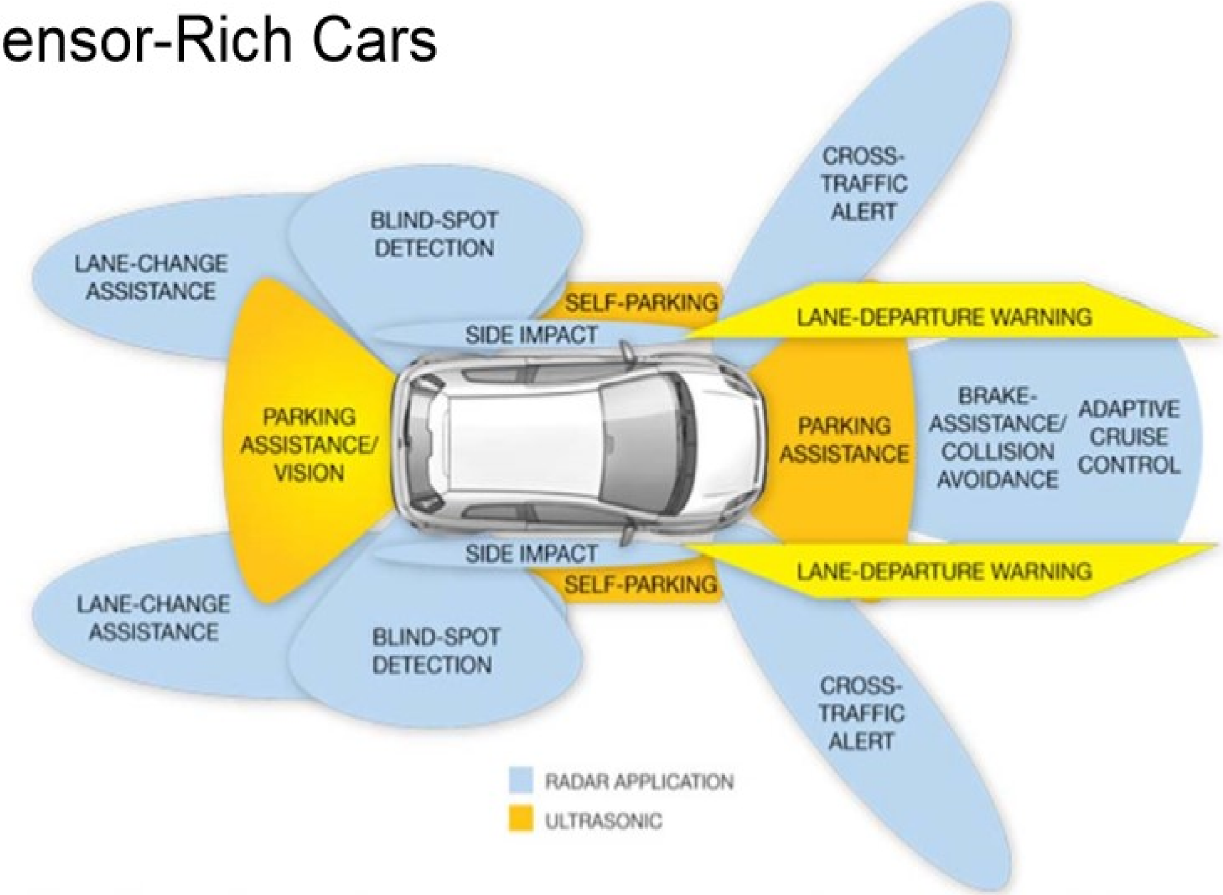
Sensors:

- Cameras
- Accelerometers
- Gyroscopes
- Strain gauges
- Microphones
- Magnetometers
- Radar/Lidar
- Chemical sensors
- Pressure sensors
- Switches

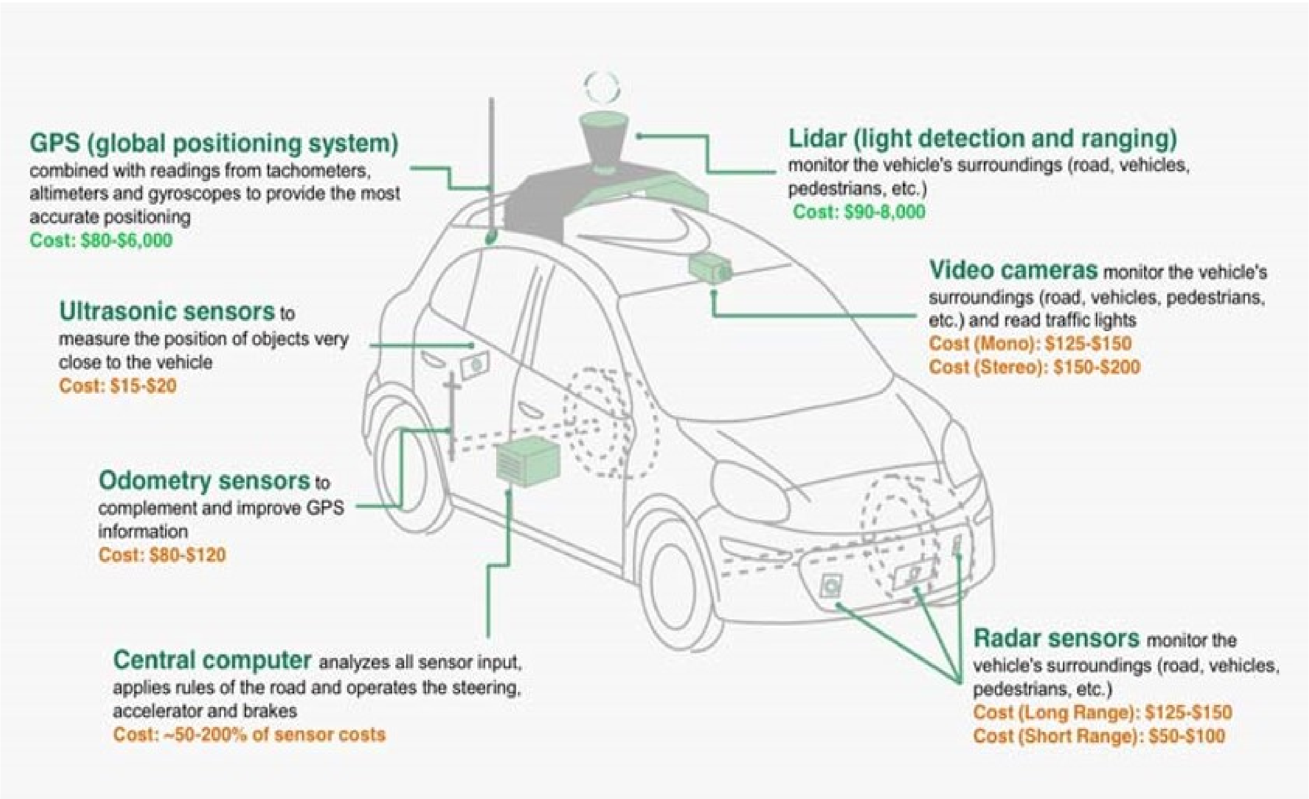
Actuators:

- Motor controllers
- Solenoids
- LEDs, lasers
- LCD and plasma displays
- Loudspeakers
- Switches
- Valves

Sensor-Rich Cars



Sensor-Rich Cars



GPS (global positioning system) combined with readings from tachometers, altimeters and gyroscopes to provide the most accurate positioning
Cost: $80-$6,000

Lidar (light detection and ranging) monitor the vehicle's surroundings (road, vehicles, pedestrians, etc.)
Cost: $90-8,000

Video cameras monitor the vehicle's surroundings (road, vehicles, pedestrians, etc.) and read traffic lights
Cost (Mono): $125-$150
Cost (Stereo): $150-$200

Ultrasonic sensors to measure the position of objects very close to the vehicle
Cost: $15-$20

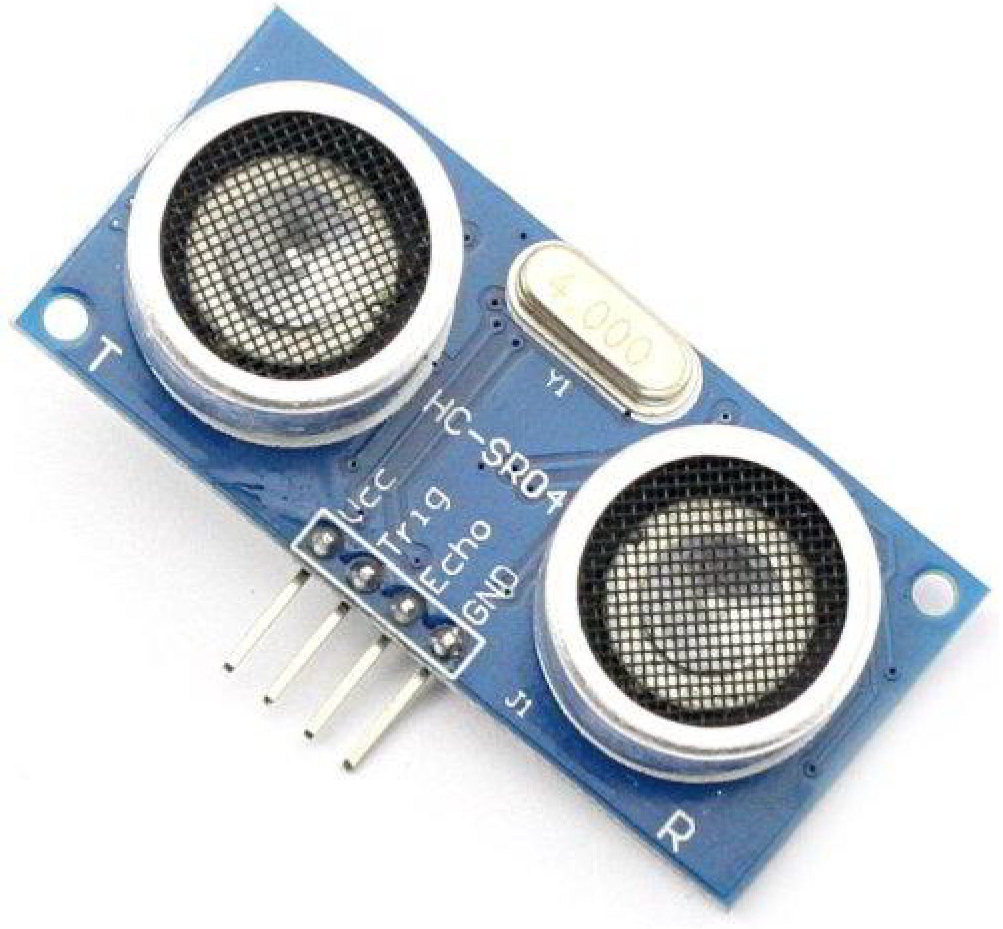
Odometry sensors to complement and improve GPS information
Cost: $80-$120

Central computer analyzes all sensor input, applies rules of the road and operates the steering, accelerator and brakes
Cost: ~50-200% of sensor costs

Radar sensors monitor the vehicle's surroundings (road, vehicles, pedestrians, etc.)
Cost (Long Range): $125-$150
Cost (Short Range): $50-$100

Ultrasonic Sensor - Interfacing with 8051

Introduction



Ultrasonic HC-SR04 Module

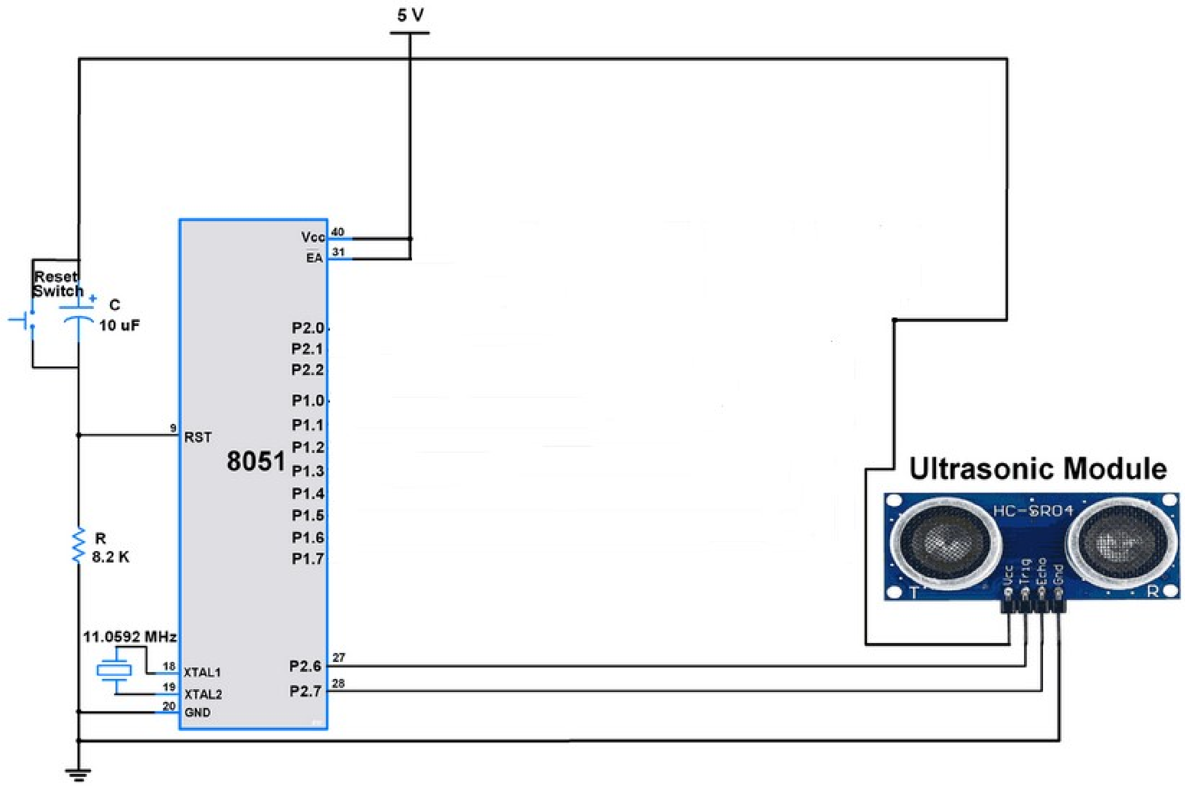
Ultrasonic Module HC-SR04 works on the principle of SONAR and RADAR system.

- The HC-SR04 module has an ultrasonic transmitter, receiver, and control circuit on a single board.
- The module has only 4 pins, Vcc, Gnd, Trig, and Echo.
- When a pulse of 10µsec or more is given to the Trig pin, 8 pulses of 40 kHz are generated. After this, the Echo pin is made high by the control circuit in the module.
- The echo pin remains high till it gets the echo signal of the transmitted pulses back.

- The time for which the echo pin remains high, i.e. the width of the Echo pin gives the time taken for generated ultrasonic sound to travel towards the object and return.
- Using this time and the speed of sound in air, we can find the distance of the object using a simple formula for distance using speed and time.

For more information about ultrasonic module HC-SR04 and how to use it, refer to the topic Ultrasonic Module HC-SR04 in the sensors and modules section.

Interfacing Diagram



HC-SR04 Ultrasonic Module Interfacing with 8051

Example

Here let’s design an application in which we will find a distance to an object by interfacing ultrasonic module HC-SR04 with 8051(here AT89S52 used) and display the distance on 16x2 LCD.

Steps of Programming

1. 8051 microcontroller needs to transmit at least 10 us trigger pulse to the HC-SR04 Trig Pin.
2. After getting a trigger pulse, HC-SR04 automatically sends eight 40 kHz sound waves and waits for rising edge output at the Echo pin.
3. When the rising edge capture occurs at the Echo pin which is connected to the input of 8051, start Timer of 8051 and again wait for the falling edge on the Echo pin.
4. As soon as the falling edge is captured at the Echo pin, the microcontroller reads the count of the Timer. This time count is used to calculate the distance to an object.

Calculation (distance in cm)

$$Distance = \frac{SoundVelocity * Time}{2}$$

Where,

Sound Velocity = 34300 (in cm per second)

Here, the oscillator frequency of AT89S52 (8051) is 11.0592 MHz then the timer frequency of 8051 will be 921.6 kHz. So, the time required to execute 1 instruction is 1.085 us.

So, the timer gets incremented after 1.085 us time elapse.

e.g.

$$= \frac{34300 * TimerCount * 1.085 * 10^{-6}}{2}$$

Applications of Ultrasonic Sensors in embedded systems

- Loop control
- Roll diameter, tension control, winding, and unwind
- Liquid level control
- Thru beam detection for high-speed counting
- Full detection
- Thread or wire break detection
- Robotic sensing
- Stacking height control
- 45° Deflection; inkwell level detection; hard to get at places
- People detection for counting
- Contouring or profiling using ultrasonic systems
- Vehicle detection for car wash and automotive assembly
- Irregular parts detection for hoppers and feeder bowls
- Presence detection
- Box sorting using a multi-transducer ultrasonic monitoring system

Advantages of ultrasonic sensors

1. The advantages of ultrasonic sensors include being unaffected by the colour of the objects being detected, including translucent or transparent objects such as water or glass.
2. Their minimum and maximum ranges are quite flexible, with most ultrasonic sensors capable of detecting as near as a few centimetres up to approximately five meters.
3. Specifically configured modules can even measure up to nearly 20 meters.
4. Ultrasonic sensors provide relatively precise measurements, errors usually within 1%, and even more precision if desired.
5. They can make many measurements per second, yielding quick refresh rates.
6. They are usually quite cheap because they do not need rare materials.
7. Moreover, ultrasonic sensors can resist electrically noisy environments and most acoustic noise, particularly when using modules equipped with encoded sound waves.

Disadvantages of ultrasonic sensors

1. As the speed of sound is dependent on temperature and humidity, ambient conditions may change the precision of the measurements.
2. Although the detection area is three dimensional, an ultrasonic sensor only detects that there is something a certain distance from the detector and cannot provide feedback on where the object is in the sensing area nor any features such as shape or colour.
3. They are relatively small in size and can be easily integrated into cars or industrial applications, but they may be too large for very small embedded projects.
4. Like any sensor, they may get dirty, wet, or frozen, which will cause them to be erratic or non-functional.
5. Due to their dependence on sound, which in turn depends on certain medium, ultrasonic sensors cannot work in a vacuum.

****

Actuator - DC motor interfacing with 8051

Small DC Motor

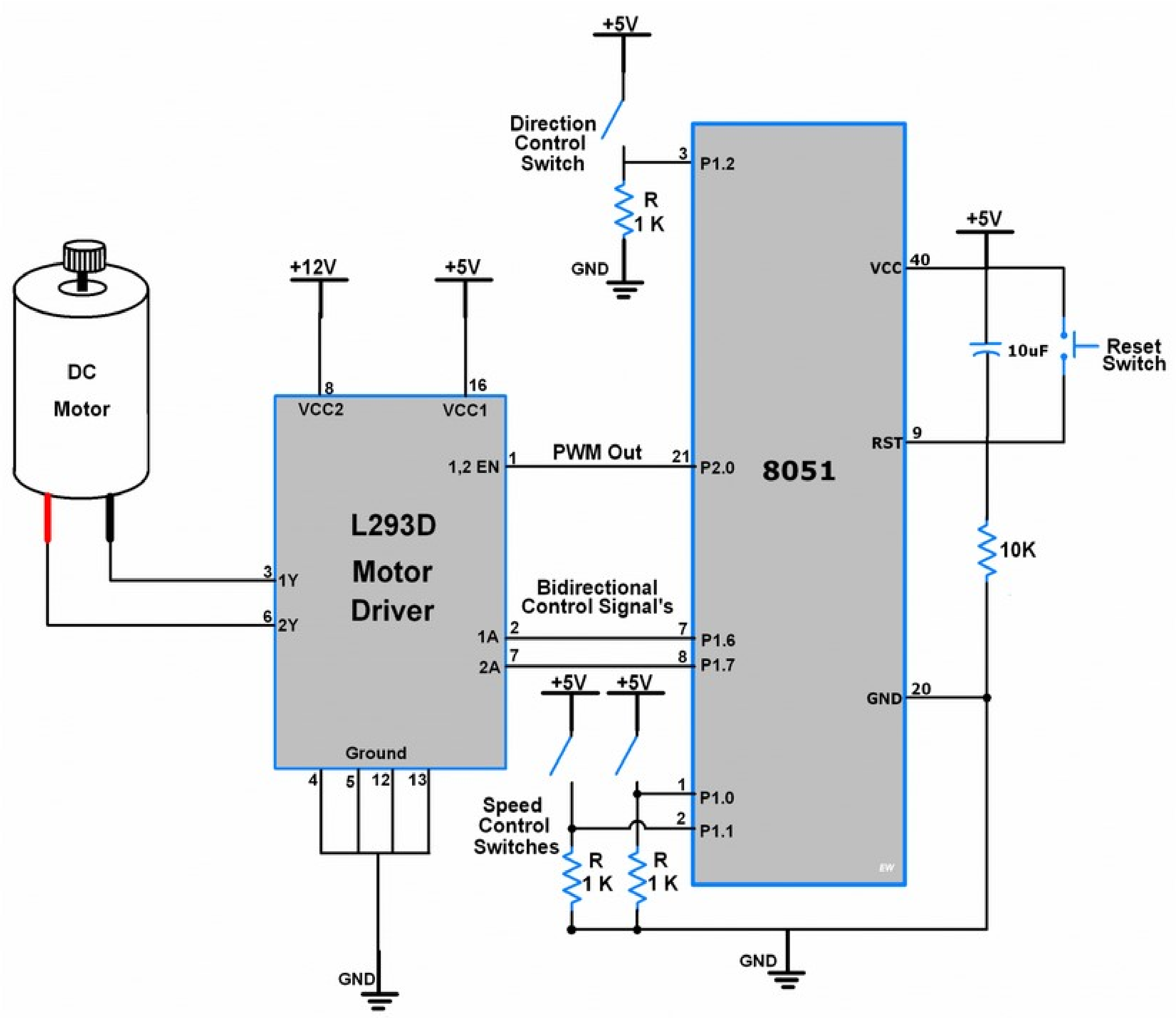
Small DC Motor

DC motor converts electrical energy in the form of Direct Current into mechanical energy.

- In the case of the motor, the mechanical energy produced is in the form of a rotational movement of the motor shaft.
- The direction of rotation of the shaft of the motor can be reversed by reversing the direction of Direct Current through the motor.

- The motor can be rotated at a certain speed by applying a fixed voltage to it. If the voltage varies, the speed of the motor varies.
- Thus, the DC motor speed can be controlled by applying varying DC voltage; whereas the direction of rotation of the motor can be changed by reversing the direction of current through it.
- For applying varying voltage, we can make use of the PWM technique.
- For reversing the current, we can make use of an H-Bridge circuit or motor driver ICs that employ the H-Bridge technique or any other mechanisms.

DC Motor Connection to 8051 using Motor Driver



8051 DC Motor Connection using L293D Motor Driver

Let’s interface the DC motor with the AT89S52 microcontroller and control the DC motor speed by using the Speed Increment Switch and Speed Decrement Switch connected to the Microcontroller port and direction by using Direction Switch.

We are going to use the L293D motor driver IC to control the DC motor movement in both directions. It has an in-built H-bridge motor drive.

- As shown in the above figure we have connected two toggle switches on the P1.0 and P1.1 pin of the AT89S52 microcontroller to change the speed of the DC motor by 10%.
- One toggle switch at pin P1.2 controls the motor's rotating direction.
- P1.6 and P1.7 pins are used as output direction control pins. It provides control to motor1 input pins of the L293D motor driver which rotate the motor clockwise and anticlockwise by changing their terminal polarity.
- And Speed of the DC Motor is varied through PWM Out pin P2.0.
- Here we are using the timer of AT89S52 to generate PWM.

Applications of DC Series motor

DC series motor is suitable for both high and low power drives, for fixed and variable speed electric drives. This type of motor has simple construction. Also, it is easy for design and maintenance. Because of its high starting torque, this motor uses in the cheap toys and automotive applications such as,

- Cranes
- Air compressor
- Lifts
- Elevators
- Winching system
- Hair drier
- Vacuum cleaner and in speed regulation application
- power tools
- Sewing machine

Applications of DC Shunt motor

DC shunt motor provides the constant speed. This type of motor mostly uses in the constant speed application from no load to full load. The applications are,

- Wiper
- Automatic windscreen
- Drills
- Conveyors
- Fans
- Boring mills
- Shapers
- Blowers
- Spinning and weighing machine
- Centrifugal pumps