Overview

Micro-controllers are useful to the extent that they communicate with other devices, such as sensors, motors, switches, keypads, displays, memory and even other micro-controllers.

Many interface methods have been developed over the years to solve the complex problem of balancing circuit design criteria such as features, cost, size, weight, power consumption, reliability, availability, manufacturability.

Many microcontroller designs typically mix multiple interfacing methods. In a very simplistic form, a micro-controller system can be viewed as a system that reads from (monitors) inputs, performs processing and writes to (controls) outputs.
Digital Inputs/Outputs

On/OFF control and monitoring.

Advantages
- Simplest interface
- Lowest-cost to implement (built into the microcontroller)
- High speed
- Low programming overhead

Disadvantages
- Only on/off control/monitoring
- Short distance, few feet maximum.
- Single device control/monitoring

Digital Input Example: Reading the status of buttons or switches

Single-ended (non-matrix) switches

8051 Microcontroller (AT89C51ED2)

Advantages
- Simplest interface
- Lowest-cost to implement (built into the microcontroller)
- High speed
- Low programming overhead

Disadvantages
- Only on/off control/monitoring
- Short distance, few feet maximum.
- Single device control/monitoring

Digital Input Example: Keypad Interface

4X4 Matrix Keypad Interface

8051 Microcontroller (AT89C51ED2)

Columns

 Rows
Digital Output Example: LED control

LED Interface

8051 Microcontroller (AT89C51ED2)

P0.0
P0.1
P0.2
P0.3

Current Limiting Resistors

LED's

Digital Output Example: Relay control

Relay Interface

8051 Microcontroller (AT89C51ED2)

P1.4

7407
Analog Inputs/Outputs

Voltage-based control and monitoring.

**Voltage type:** Typical ranges
- 0 to 2.5V
- 0 to 4V
- 0 to 5V
- +/- 2.5V
- +/- 4V
- +/- 5V

**Current type:** Typical ranges
- 0-20mA
- 4-20mA

**Advantages**
- Simple interface
- Low cost for low-resolutions
- High speed
- Low programming overhead

**Disadvantages**
- High cost for higher resolutions
- Not all microcontrollers have analog inputs/outputs built-in
- Complicates the circuit design when external ADC or DAC are needed.
- Short distance, few feet maximum.

**Analog Interface**

```
Strain-gage

Potentiometer

Vcc

LM35 Temperature Sensor

Vcc

8051 Microcontroller (AT89C51ED2)

Digital/Analog Converter (DAC)

4-20mA Output

Amplifier
```
Parallel Bus

Consists of multiple digital inputs/outputs. Most common types:

- 4-bit
- 8-bit (e.g. Centronics)
- 16-bit (e.g. ISA)
- 32-bit (e.g. PCI)

**Advantages**
- High speed
- High throughput: Several bits are transmitted on one clock transition
- Low cost

**Disadvantages**
- Large number of microcontroller pins that needed for implementing the parallel bus

**Example:** LCD Interface

4-bit LCD Interface

8051 Microcontroller (AT89C51ED2)

<table>
<thead>
<tr>
<th>P0.7</th>
<th>P0.6</th>
<th>P0.5</th>
<th>P0.4</th>
<th>P0.3</th>
<th>P0.2</th>
<th>P0.1</th>
<th>P0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alphanumeric LCD

<table>
<thead>
<tr>
<th>07</th>
<th>06</th>
<th>05</th>
<th>04</th>
<th>03</th>
<th>02</th>
<th>01</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>R/W</th>
<th>RS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hello World

8-bit LCD Interface

8051 Microcontroller (AT89C51ED2)

<table>
<thead>
<tr>
<th>P0.7</th>
<th>P0.6</th>
<th>P0.5</th>
<th>P0.4</th>
<th>P0.3</th>
<th>P0.2</th>
<th>P0.1</th>
<th>P0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alphanumeric LCD

<table>
<thead>
<tr>
<th>07</th>
<th>06</th>
<th>05</th>
<th>04</th>
<th>03</th>
<th>02</th>
<th>01</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>R/W</th>
<th>RS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hello World
Serial Buses

I2C (Inter Integrated Circuit bus)

2-wire interface with one master and multiple slaves (multi-master configurations possible). Originated by Philips Semiconductor in the early 80’s to connect a microcontroller to peripheral devices in TV sets.

Signals: DATA (SDA), CLOCK (SCL) and Ground. SDA is always bi-directional; SCL is bi-directional only in multi-master mode.

Maximum allowable capacitance on the lines is 400 pF. Typical device capacitance is 10 pF.

To start the communications, the bus master (typically a microcontroller) places the address of the device with which it intends to communicate (the slave) on the bus. All slave devices monitor the bus to determine if the master device is sending their address. Only the device with the correct address communicates with the master.

By definition, I2C is 5V.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Multiple slave devices can be accessed with</td>
<td>• Short distance</td>
</tr>
<tr>
<td>only 3 wires</td>
<td>• Slow speed: 100 KHz although 400 KHz</td>
</tr>
<tr>
<td>• Low-cost to implement</td>
<td>and 1 MHz slave devise exist. These</td>
</tr>
<tr>
<td>• Implemented in hardware or software</td>
<td>can not coexist with slower devices.</td>
</tr>
<tr>
<td>• Ease to implement, many examples</td>
<td>• Limited device addresses</td>
</tr>
<tr>
<td>• Supports multi-master configuration</td>
<td></td>
</tr>
</tbody>
</table>

2-wire (I2C) interface

- 8051 Microcontroller (AT89C51ED2)
- AT24C04 EEPROM
- DS1307 Real-Time Clock
- Lithium Battery
- Crystal
**Start and Stop**

An I2C master prepares to communicate with a slave device first by generating a Start condition on the bus. Start condition is defined as SDA signal going low while SCL signal is high. Stop condition is defined as SDA going high while SCL is high.

![Start and Stop Diagram](image)

**Data Validity**

Data can change while the clock is low. Data should remain stable while the clock is going high.

![Data Validity Diagram](image)

**Acknowledge (ACK)**

![Acknowledge (ACK) Diagram](image)
Writing a byte to a serial EEPROM (24C04) on the I2C bus:

where Device Address is defined as

2K

<table>
<thead>
<tr>
<th>A2</th>
<th>A1</th>
<th>A0</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>MSB</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>LSB</td>
</tr>
</tbody>
</table>

4K

<table>
<thead>
<tr>
<th>A2</th>
<th>A1</th>
<th>P0</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>MSB</td>
</tr>
</tbody>
</table>

8K

<table>
<thead>
<tr>
<th>A2</th>
<th>P1</th>
<th>P0</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>MSB</td>
</tr>
</tbody>
</table>

16K

<table>
<thead>
<tr>
<th>P2</th>
<th>P1</th>
<th>P0</th>
<th>R/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>MSB</td>
</tr>
</tbody>
</table>

P0, P1, P2 indicate the page number (2Kbit pages). A0, A1, A2 indicate the device number on the bus.

Reading a byte from a serial EEPROM (24C04) on the I2C bus (starting from the current address)
SPI (Serial Peripheral Interface)

4-wire interface with one master and multiple slaves. Signals: DATA IN, DATA OUT, CLOCK, CS (Chip Select)

Originated by Motorola, SPI bus is a relatively simple synchronous serial interface for connecting low speed external devices using minimal number of wires. A synchronous clock shifts serial data into and out of the microcontrollers in blocks of 8 bits.

SPI bus is a master/slave interface. Whenever two devices communicate, one is referred to as the "master" and the other as the "slave" device. The master drives the serial clock. SPI is full duplex: Data is simultaneously transmitted and received.

Example: Multimedia Card (MMC) Interface using SPI

**Advantages**
- Multiple slave devices can be accessed with only few wires
- Low-cost
- Implemented in hardware or software
- Ease to implement, many examples
- Can be high speed (e.g. 4MHz or higher if implemented in hardware)

**Disadvantages**
- Short distance
- Data and clock lines can be shared but each device requires a separate Chip Select signal, limiting the number of devices in limited I/O systems

MMC Interface

- 8051 Microcontroller (AT89C51ED2)
  - P3.5
  - P3.7
  - P3.3
  - P3.4
- Multimedia Card
  - MMC
  - CS
  - DIN
  - DOUT
  - CLK
1-wire

Originated by Dallas Semiconductor (now part of MAXIM) to address a variety of peripherals, sensors, and memory chips from a single wire interface (DATA and Ground). One signal wire carries both operating power and signal. Usually the network is built using a wire pair where one wire carries the signal and power and the other wire is ground. The system is sensitive to the right timing to operate well.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple slave devices can be accessed with only 2 wires</td>
<td>• Slow speed</td>
</tr>
<tr>
<td>Low-cost</td>
<td>• 1-wire slave devices typically has to come from one source: Dallas Semiconductor</td>
</tr>
<tr>
<td>Implemented in hardware or software</td>
<td></td>
</tr>
<tr>
<td>Ease to implement, many examples</td>
<td></td>
</tr>
<tr>
<td>Relatively long distance. Theoretically 300 meters but this is limited in</td>
<td></td>
</tr>
<tr>
<td>practice due to noise and cable capacitance</td>
<td></td>
</tr>
</tbody>
</table>

RS232
Asynchronous communications

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Popular interface with many examples</td>
<td>• More suitable for system to system communications, not so much for chip to chip or chip to sensor</td>
</tr>
<tr>
<td>• Many compatible legacy devices</td>
<td>• Low speed for long distance, 115200 baud can be achieved with small microcontrollers using short distances</td>
</tr>
<tr>
<td>• Relatively long distance, 50 feet maximum for low baud rates although longer distances work in practice, with low baud rates and error correction</td>
<td>• Requires transceiver chips which add to system cost (TTL/CMOS level RS232 can be used without transceiver chips)</td>
</tr>
<tr>
<td>• Immune to noise due to +/-5 Volts or higher voltage levels for logic &quot;0&quot; and &quot;1&quot;</td>
<td>• Single master/single slave</td>
</tr>
<tr>
<td>• Implemented in hardware or software</td>
<td></td>
</tr>
<tr>
<td>• Ease to implement, many examples</td>
<td></td>
</tr>
</tbody>
</table>

• Single master/single slave
RS485

Asynchronous communications

**Advantages**
- Popular interface with many examples
- Very long distance, thousands of feet
- Immune to noise due to differential voltage
- Implemented in hardware or software
- Ease to implement, many examples
- Widely used in industrial automation
- Higher speeds beyond 115200 baud

**Disadvantages**
- More suitable for system to system communications, not so much for chip to chip or chip to sensor
- Requires transceiver chips and twisted pair cable with terminating resistors which add to system cost.

RS485 Network Topology: Any station can communicate with any other station, but not at the same time.

Correct termination resistor that matches the characteristic impedance of the cable is very important in RS485. Otherwise, reflected waves will result in distortions of the original waveform to the point where data errors occur.
Ethernet

Advantages

• Very high speed (10Mbit to 100Mbit/s)
• Very long distance, hundreds of feet can be achieved, more with hubs and switches
• Immune to noise
• Widely used in industrial automation due to noise immunity

Disadvantages

• Cost
• More suitable for system to system communications, not so much for chip to chip/sensor
• Requires Ethernet chipset, transformer, jack and special cabling that add to system cost.
• Complicated to implement
• High code footprint

10 MBit ETHERNET NETWORKING WITH MINI-MAX/51-E

Operator Terminal Station 1

MINI-MAX/51-E
IP: 192.168.0.100

Operator Terminal Station 2

MINI-MAX/51-E
IP: 192.168.0.101

Product Monitoring Station 3

MINI-MAX/51-E
IP: 192.168.0.102

10 Mbit Ethernet

Supervisory Control and Data Acquisition PC
IP: 192.168.0.2
<table>
<thead>
<tr>
<th>Network Concept</th>
<th>I²C™</th>
<th>SMBUS™</th>
<th>SPI™</th>
<th>MicroWire/PLUS™</th>
<th>M-Bus (ETN1434)</th>
<th>CAN (ISO11898)</th>
<th>LIN Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Size</td>
<td>Single master, multiple slaves</td>
<td>Multiple masters, multiple slaves</td>
<td>Multiple masters, multiple slaves</td>
<td>Single master, multiple slaves</td>
<td>Multiple masters, multiple slaves</td>
<td>Multiple masters, multiple slaves</td>
<td>Single master, multiple slaves</td>
</tr>
<tr>
<td>Optional signals</td>
<td>N/A</td>
<td>SMBUS#</td>
<td>N/A</td>
<td>N/A</td>
<td>2nd GND, Power, Shield</td>
<td>2nd GND, Power, Shield</td>
<td>N/A</td>
</tr>
<tr>
<td>Network Interface</td>
<td>Open drain, resistive or active master pull-up</td>
<td>Open drain, resistive or active master pull-up</td>
<td>Push-pull with tristate</td>
<td>Push-pull with tristate</td>
<td>M to S: voltage drive S to M: current load</td>
<td>Differential open drain/source or open coll/emitter</td>
<td>Open drain, resistive master pull-up</td>
</tr>
<tr>
<td>Network Voltage</td>
<td>From 2.8 to 6.0V, device specific</td>
<td>From 1.8 to 5.5V, device specific</td>
<td>From 1.8V to 5.5V, device specific</td>
<td>From 1.8V to 5.5V, device specific</td>
<td>~40V</td>
<td>VDD-VD (diode drop); ~4.8V max.</td>
<td>8 to 18V</td>
</tr>
<tr>
<td>Logic Thresholds</td>
<td>Vary with network voltage</td>
<td>Fixed level: &gt;1.5V, &gt;3.0V VDD-related level: &lt;30%, &gt;70% of VDD</td>
<td>&lt;0.8V, &gt;2.1V</td>
<td>VDD-related level: &lt;20% (30%), &gt;70% of VDD (inconsistent)</td>
<td>Master to slave: 24V, 36V nominal Slave to master: &lt;1.5mA, &gt;11mA</td>
<td>Master to slave: &lt;1.5V (dominant); driver specification</td>
<td>VDD-related level: &lt;20%, &gt;80% of VDD (receiver spec.)</td>
</tr>
<tr>
<td>Transmission</td>
<td>LS bit first, half-duplex</td>
<td>MS bit first plus Acknowledge bit, half-duplex</td>
<td>MS bit first plus Acknowledge bit, half-duplex</td>
<td>MS bit first, full-duplex</td>
<td>LS bit first, half-duplex, acknowledge response</td>
<td>MS bit first, half-duplex</td>
<td>LS bit first, half-duplex</td>
</tr>
<tr>
<td>Address Format</td>
<td>56 bits</td>
<td>7 bits, (10 bits defined but not implemented)</td>
<td>7 bits, (10 bits defined but not implemented)</td>
<td>N/A</td>
<td>8 bits (primary address), 64 bits (secondary address)</td>
<td>8 bits (primary address), 29 bits (extended format)</td>
<td>Message identifier 8 bits, including 2 parity bits</td>
</tr>
<tr>
<td>Network Inventory</td>
<td>Automatic, supports dynamic topology change</td>
<td>Automatic, supports dynamic topology change</td>
<td>ARP, Address Resolution Protocol (Rev. 2.0 only)</td>
<td>N/A; slave select (CS) hard-coded in firmware</td>
<td>Automatic</td>
<td>N/A; message-based protocol, not address based</td>
<td>N/A; message-based protocol, not address based</td>
</tr>
<tr>
<td>Gross Data Rate</td>
<td>Standard: ~0 to 16.3k bps (Overdrive: ~0 to 142k bps)</td>
<td>Standard: ~0 to 100k bps; Fast: ~0 to 200k bps; High-Speed: ~0 to 3.4M bps</td>
<td>10k to 100k bps</td>
<td>~0 to ~10 M bps (device specific)</td>
<td>300, 2400, 9600 bps</td>
<td>~0 to 1M bps</td>
<td>~1k to ~20k bps</td>
</tr>
<tr>
<td>Access Time</td>
<td>Standard: ~5.4ms Overdrive: ~0.6ms at maximum speed</td>
<td>Standard: ~95µs Fast: ~23µs at maximum speed</td>
<td>~95µs @ 10k bps</td>
<td>N/A</td>
<td>Primary address, 2400 bps: 13.75ms (short frame), 27.5ms (long frame)</td>
<td>At 1M bps 19µs (standard) or 39µs (extended) from start of frame to 1st data bit</td>
<td>At 20k bps 1.7ms from start of frame to 1st data bit</td>
</tr>
<tr>
<td>Data Protection</td>
<td>8-bit and 16-bit CRC</td>
<td>PEC Packet Error Code (Rev.1.1, 2.0)</td>
<td>N/A</td>
<td>N/A</td>
<td>Even parity, check sum, frames</td>
<td>15-bit CRC, frames, frame acknowledge</td>
<td>Check sum, frames</td>
</tr>
<tr>
<td>Collision Detection</td>
<td>Yes, through non-matching CRC</td>
<td>Yes (multi-master operation only)</td>
<td>Yes (Rev. 2.0 only)</td>
<td>N/A</td>
<td>Yes (“medium” and “strong” collisions)</td>
<td>Yes: CSMA/CD</td>
<td>Yes, through check sum</td>
</tr>
<tr>
<td>Slave supply</td>
<td>Parasitic (I²C only)</td>
<td>VDD only</td>
<td>VDD only</td>
<td>VDD only</td>
<td>VDD only</td>
<td>Parasic and/or local</td>
<td>VDD only local or Parasitic only</td>
</tr>
<tr>
<td>$V_{DD}$ (exception)</td>
<td>supply</td>
<td>remote source</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>--------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>