Analog to Digital Converters / Digital to Analog Converters, 7-Segments & LCDs

Updated: 4/9/18
Data Converters: Basic Concepts

- Analog signals are continuous, with infinite values in a given range.
- Digital signals have discrete values such as on/off or 0/1.
- Limitations of analog signals
  - Analog signals pick up noise as they are being amplified.
  - Analog signals are difficult to store.
  - Analog systems are more expensive in relation to digital systems.
Data Converters: Basic Concepts

- Advantages of digital systems (signals)
  - Noise can be reduced by converting analog signals in 0s and 1s.
  - Binary signals of 0s/1s can be easily stored in memory.
  - Technology for fabricating digital systems has become so advanced that they can be produced at low cost.

- The major limitation of a digital system is how accurately it represents the analog signals after conversion.

Accuracy is the key concept!
Embedded System

- A typical system that converts signals from analog to digital and back to analog includes:
  - A transducer that converts non-electrical signals into electrical signals
  - An A/D converter that converts analog signals into digital signals
  - A digital processor that processes digital data (signals)
  - A D/A converter that converts digital signals into equivalent analog signals
  - A transducer that converts electrical signals into real life non-electrical signals (sound, pressure, and video)

So, how are ADC different from one another and how do they work?
# ADC Comparison

<table>
<thead>
<tr>
<th>Product Attribute</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer:</td>
<td>Analog Devices Inc.</td>
</tr>
<tr>
<td>Product Category:</td>
<td>Analog to Digital Converters - ADC</td>
</tr>
<tr>
<td>RoHS:</td>
<td><img src="https://www.mouser.com/ProductDetail/Analog-Devices-Inc/AD7641/?qs=%2Fjp%3D0%26q%3Df%26id%3D0%26gclid%3DCjwKCAjwwbHBBBWEiwAAM1V7EK9LzpTChOykFAN9d0pNLQfmaODQEKiu_nXcm73phAQVnWd7WwR0CnunQQA/s_1wE" alt="RoHS Details" /></td>
</tr>
<tr>
<td>Series:</td>
<td>AD7641</td>
</tr>
<tr>
<td>Mounting Style:</td>
<td>SMD/SMT</td>
</tr>
<tr>
<td>Package / Case:</td>
<td>LFCSP-48</td>
</tr>
<tr>
<td>Resolution:</td>
<td>18 bit</td>
</tr>
<tr>
<td>Number of Channels:</td>
<td>1 Channel</td>
</tr>
<tr>
<td>Sampling Rate:</td>
<td>2 MS/s</td>
</tr>
<tr>
<td>Input Type:</td>
<td>Differential</td>
</tr>
<tr>
<td>Interface Type:</td>
<td>Parallel, Serial, 2-Wire, DSP, SPI</td>
</tr>
<tr>
<td>Architecture:</td>
<td>SAR</td>
</tr>
<tr>
<td>Reference Type:</td>
<td>External, Internal</td>
</tr>
<tr>
<td>SNR - Signal to Noise Ratio:</td>
<td>93.5 dB</td>
</tr>
<tr>
<td>Minimum Operating Temperature:</td>
<td>- 40 °C</td>
</tr>
<tr>
<td>Maximum Operating Temperature:</td>
<td>+ 85 °C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Attribute</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer:</td>
<td>Texas Instruments</td>
</tr>
<tr>
<td>Product Category:</td>
<td>Analog to Digital Converters - ADC</td>
</tr>
<tr>
<td>RoHS:</td>
<td><img src="https://www.mouser.com/ProductDetail/Analog-Devices-Inc/AD7641/?qs=%2Fjp%3D0%26q%3Df%26id%3D0%26gclid%3DCjwKCAjwwbHBBBWEiwAAM1V7EK9LzpTChOykFAN9d0pNLQfmaODQEKiu_nXcm73phAQVnWd7WwR0CnunQQA/s_1wE" alt="RoHS Details" /></td>
</tr>
<tr>
<td>Series:</td>
<td>TLV0832</td>
</tr>
<tr>
<td>Mounting Style:</td>
<td>Through Hole</td>
</tr>
<tr>
<td>Package / Case:</td>
<td>PDIP-8</td>
</tr>
<tr>
<td>Resolution:</td>
<td>8 bit</td>
</tr>
<tr>
<td>Number of Channels:</td>
<td>2 Channel</td>
</tr>
<tr>
<td>Sampling Rate:</td>
<td>44.7 kS/s</td>
</tr>
<tr>
<td>Input Type:</td>
<td>Differential/Single-Ended</td>
</tr>
<tr>
<td>Interface Type:</td>
<td>Serial</td>
</tr>
<tr>
<td>Architecture:</td>
<td>SAR</td>
</tr>
<tr>
<td>Reference Type:</td>
<td>Supply</td>
</tr>
<tr>
<td>Analog Supply Voltage:</td>
<td>2.7 V to 3.6 V</td>
</tr>
<tr>
<td>Digital Supply Voltage:</td>
<td>2.7 V to 3.6 V</td>
</tr>
<tr>
<td>Minimum Operating Temperature:</td>
<td>- 40 °C</td>
</tr>
<tr>
<td>Maximum Operating Temperature:</td>
<td>+ 85 °C</td>
</tr>
</tbody>
</table>
A/D Converter: Requires Three Steps

- In order to change an analog signal to digital, the input analog signal is **sampled** at a high rate of speed.
- The amplitude at each of those sampled moments is converted into a number equivalent – this is called **quantization**.
- These numbers are simply the combinations of the 0s and 1s used in computer language – this called **encoding**.

http://www.cybercollege.com/tvp008.htm
A/D Conversion:
Pulse Code Modulation/Demodulation

![Diagram of A/D Conversion](image)

- **Modulation**
  - Original Signal
  - Analog-to-Digital Conversion
  - Analog Signal
  - Filter
  - Sample
  - Quantize
  - Encode
  - 8-Bit Binary Number
- **Demodulation**
  - PCM Signal
  - Filter
  - PAM Signals
  - Decoder
  - Original Signal
Analog-to-Digital Conversion Accuracy

- A simple hypothetical A/D converter circuit with one analog input signal and three digital output lines with eight possible binary combinations: 000 to 111

- Important points:
  - **Maximum value** this quantization process reaches is $\frac{7}{8}$ V for a 1 V analog signal; includes $\frac{1}{8}$ V an inherent error
  - $\frac{1}{8}$ V (an inherent error) is also equal to the value of the Least Significant Bit (LSB) = 001.
  - **Resolution** of a converter is defined in terms of the number of discrete values it can produce; also expressed in the number of bits used for conversion or as $\frac{1}{2^n}$ where $n =$ number of bits
  - The value of the most significant bit (MSB) -100- is equal to $\frac{1}{2}$ the voltage of the full-scale value of 1 V.
  - The value of the largest digital number 111 is equal to full-scale value minus the value of the LSB.
  - The **quantization error** can be reduced or the resolution can be improved by increasing the number of bits used for the conversion
ADC Components

- Op-amps
  - Comparators & Schmitt triggers
- Switches
- Counters
A little Detour: Opamp Review

- [http://www.engin.brown.edu/courses/en123/Lectures/DAconv.htm](http://www.engin.brown.edu/courses/en123/Lectures/DAconv.htm)
- Check out the Applet on my web
Opamps

- Ideal opamps
  - Infinite BW
  - Infinite voltage gain
  - Infinite input impedance
  - Zero output impedance

- Practical opamps
  - Wide BW
  - Very high voltage gain
  - Very high input impedance
  - Very low output impedance

http://www.chem.uoa.gr/Applets/AppletOpAmps/Appl_OpAmps2.html
Closed Loop Frequency Response

- **Non-inverting**
  - Source is connected to the non-inverting input
  - Feedback is connected to the inverting input
  - If $R_f$ and $R_i$ are zero, then unity feedback used for buffering
  - $A_v = 1 + \frac{R_f}{R_i}$

- **Inverting**
  - Feedback and source are connected to the inverting input
  - $A_v = -\frac{R_f}{R_i}$

Try this: [http://www.falstad.com/circuit/e-opamp.html](http://www.falstad.com/circuit/e-opamp.html)
Comparators

- Determines which input is larger
- A small difference between inputs results in a maximum output voltage (high gain)
- Zero-level detection
- Non-zero-level detection
Example

\[ V_{\text{ref}} = \frac{V_{\text{in}}(\text{max}).R_2}{R_1+R_2} = 1.63 \text{ V} \]

http://www.falstad.com/circuit/e-opamp.html
Back to A/D Converters....
A/D Conversion – Categories

- Can be classified in four groups:
  - **Flash**: uses multiple comparators in parallel.
    - The known signal is connected to one side of the comparator and the analog signal to be converted to the other side of the comparator.
    - The output of the comparators provides the digital value.
    - This is a high-speed, high cost converter.
  - **Integrator**:
    - Charges a capacitor for a given amount of time using the analog signal.
    - It discharges back to zero with a known voltage and the counter provides the value of the unknown signal.
    - Provides slow conversion but low noise.
    - Often used in monitoring devices (e.g., voltmeters)
  - **Delta Sigma**
  - **Successive approximation converter (integrator type)**
A/D Conversion –
Categories – Bit Resolution and Speed

- Integrator:
  - Charges a capacitor for a given amount of time using the analog signal.
  - It discharges back to zero with a known voltage and the counter provides the value of the unknown signal.
  - Provides slow conversion but low noise.
  - Often used in monitoring devices (e.g., voltmeters)

- Flash:
  - Uses multiple comparators in parallel.
  - The known signal is connected to one side of the comparator and the analog signal to be converted to the other side of the comparator.
  - The output of the comparators provides the digital value.
  - This is a high-speed, high-cost converter.

- Delta Sigma

- Successive Approximation

- Sigma-Delta

The diagram shows a comparison of the maximum conversion speed for different methods (Sigma-Delta, Successive Approximation, Flash) across different resolutions (8, 16, 24 bits). The graph on the left uses a linear scale for maximum conversion speed, while the graph on the right uses a log scale.
A/D Conversion

- **Flash Converter**
  - The circuit consists of 4 comparators whose inverting inputs are connected to a voltage divider.
  - A comparator is basically an operational amplifier used without feedback.
  - The outputs of the comparators correspond to a digital word.
  - When the input rises above \( V_{n1} \), the first comparator will switch to a high output voltage causing the LED to light up, indicating a (0001).
  - For larger input voltages the output of other comparators will switch high as well. For large input voltages (above \( V_{n3} \)) all comparators will be high corresponding to (1111) digital output.
A/D Conversion

- **Successive approximation**: Includes a D/A (digital to analog) converter and a comparator. An internal analog signal is generated by turning on successive bits in the D/A converter.

- **Counter**: Similar to a successive approximation converter except that the internal analog signal is generated by a counter starting at zero and feeding it to the D/A converter.
Successive Approximation A/D Converter Circuit

- The SAR (successive approximation register) begins by turning on the MSB Bit7.
- $V_o$ of the D/A converter is compared with the analog input voltage $V_{in}$ in the comparator.
- If analog voltage is less than the digital voltage, Bit7 is turned off and Bit6 is turned on.
- If analog voltage is greater than the digital voltage, Bit7 is kept on and Bit6 is turned on.
- The process of turning bit on/off is continued until Bit0.
- Now the 8-bit input to the D/A converter represents the digital equivalent of the analog signal $V_{in}$.

Bit 7 is set: $b_7=1$
If $V_a < V_d \rightarrow b_7=0; b_6=1$
If $V_a > V_d \rightarrow b_7=1; b_6=1$

......
If $V_a < V_d \rightarrow b_7=0; \ldots b_0=1$
If $V_a > V_d \rightarrow b_7=1; \ldots b_0=1$
Done
Successive-approximation (SAR) ADC

- **Binary search** algorithm is used to gradually approach the input voltage – This is how it works:

  - The SAR (successive approximation register) begins by turning on the MSB Bit7
  - If analog voltage is greater than the digital voltage, Bit7 is kept on and Bit6 is turned on.
Minimum Sampling Rate: Nyquist–Shannon Sampling Theorem

- In order to be able to reconstruct the analog input signal, the sampling rate should be at least twice the maximum frequency component contained in the input signal.

- Sampling freq. is the same as maximum freq.
- Sampling freq. is 25 times higher
- Sampling freq. is 5 times higher
Sample and Hold Circuit

- If the input voltage to an A/D converter is changing, the digital output is likely to be unreliable and unstable. Therefore, the varying voltage source is connected to the ADC through a sample and hold circuit.

- Basic Operation (three distinct timing requirements):
  - When the switch is connected, it samples the input voltage.
  - When the switch is open, it holds the sampled voltage by charging the capacitor.
  - Acquisition time: time to charge the capacitor after the switch is open and settle the output.
  - Conversion time: total time needed from the start of a conversion (turning on the MSB in the SAR) until the end of the conversion (turning on/off Bit0 in the SAR) - TAD: conversion time per bit.

FET=Field Effect Transistor
Determining Minimum Sampling Time

\[ V_c(t) = V_{in} \times (1 - e^{-\frac{t}{T_c}}) \]

Sampling time is software programmable!

Larger sampling time \rightarrow Smaller sampling error

Slower ADC speed

Tradeoff
Programming ADC Sampling Time

HSI Clock 16 MHz

Prescaler /1, /2, /4

ADC Clock

default prescaler = 1

4 cycles
9 cycles
16 cycles
24 cycles
48 cycles
96 cycles
192 cycles
384 cycles

Selection

Sampling time

ADC sample time register (SMPR)
Successive-approximation (SAR) ADC

\[ T_{ADC} = T_{sampling} + T_{Conversion} \]

\[ T_{Conversion} = N \times T_{ADC\_Clock} \]

\( T_{sampling} \) is software configurable

Sampling time Includes acquisition time
ADC Conversion Time

\[ T_{ADC} = T_{sampling} + T_{Conversion} \]

Suppose ADCCLK = 16 MHz and Sampling time = 4 cycles

16 MHz \( \rightarrow \) 0.625x10^{-9} sec = 1 cycle

For 12-bit ADC

\[ T_{ADC} = 4 + 12 = 16 \text{ cycles} = 1\mu s \]

For 6-bit ADC

\[ T_{ADC} = 4 + 6 = 10 \text{ cycles} = 625\text{ns} \]

Remember:

\[ T_{Conversion} = N \times T_{ADC\_Clock} \]

\( T_{sampling} \) is software configurable
Importance of $V_{REF}$

- Some chips do not have external $V_{REF}$ – no external pin

Corresponds to $V_{REF}$

![Graph showing ADC output codes and voltage input (V)]
A/D Examples

- **Example 1**
  - Assumes the input analog voltage is changing between 0-5 V.
  - Using a 3-bit A/D converter draw the output as the input signal ramps from 0 to 5V.
  - Calculate the resolution.
  - What is the maximum possible voltage out? (this is called the full-scale output)
  - If the output is 011, what is the input?

- **Example 2**
  - Assumes the input analog voltage is changing between -5 to 5 V; using a 10-bit A/D converter.
  - Calculate the number of quantization levels.
  - Calculate the voltage resolution.
A/D Examples

Example 1
- Assumes the input analog voltage is changing between 0-5 V.
- Using a 3-bit A/D converter draw the output as the input signal ramps from 0 to 5V.
- Calculate the resolution in volts. $\frac{5}{2^{3}}V$
- What is the maximum possible voltage out? (this is called the full-scale output) $(5 – \text{Resolution})$
- If the output is 011, what is the input? $3 \times \frac{5}{8}V$

Example 2
- Assumes the input analog voltage is changing between -5 to 5 V; using a 10-bit A/D converter.
- Calculate the number of quantization levels. $2^{10}$
- Calculate the voltage resolution. $\frac{5-(-5)}{1024}=9.76 \text{ mV}$
Digital to Analog (D/A, DAC, or D-to-A) Conversion

- Converting discrete signals into discrete analog values that represent the magnitude of the input signal compared to a standard or reference voltage
  - The output of the DAC is discrete analog steps.
  - By increasing the resolution (number of bits), the step size is reduced, and the output approximates a continuous analog signal.
Analysis of a Ladder Network

- A resistive ladder network is a special type of series-parallel circuit.
- One form of ladder network is commonly used to scale down voltages to certain weighted values for digital-to-analog conversion
  - Called R/2R Ladder Network
- To find total resistance of a ladder network, start at the point farthest from the source and reduce the resistance in steps.
The R/R2 Ladder Network

Only Input 4 is HIGH

Only Input 3 is HIGH

Used for Digital-to-analog converter!
Examining Digital-to-Analog Conversion

For Extra credit: Change the circuit to generate this output:
Digital to Analog Conversion

- The resolution of a DAC is defined in terms of bits—the same way as in ADC.
- The values of LSB, MSB, and full-scale voltages calculated the same way as in the ADC.
- The largest input signal 111 is equivalent of 7/8 of the full-scale analog value.
D/A Converter Circuits (1 of 4)

- Can be designed using an operational amplifier and appropriate combination of resistors
- Resistors connected to data bits are in binary weighted proportion, and each is twice the value of the previous one.
- Each input signal can be connected to the op amp by turning on its switch to the reference voltage that represents logic 1.
  - If the switch is off, the input signal is logic 0.
D/A Converter Circuits (2 of 4)

- 3-bit D/A Converter Circuit

The transfer function of the summing amplifier:

\[ v_o = -(v_1/R_1 + v_2/R_2 + \ldots + v_n/R_n)R_f \]

Thus if all input resistors are equal, the output is a scaled sum of all inputs. If they are different, the output is a weighted linear sum of all inputs.

Summing amplifier

- R/2R Ladder Network for D/A Converter
D/A Converter Circuits

- If the reference voltage is 1 V, and if all switches are connected, the output current can be calculated as follows:

\[
I_o = I_T = I_1 + I_2 + I_3 = \frac{V_{REF}}{R_1} + \frac{V_{REF}}{R_2} + \frac{V_{REF}}{R_3} = \frac{V_{REF}}{1k} \left( \frac{1}{2} + \frac{1}{4} + \frac{1}{8} \right) = 0.875 \text{mA}
\]

- Output voltage

\[
V_O = -R_f I_T = -(1k) \times (0.875 \text{ mA}) = -0.875 \text{ V} = \left\lceil \frac{7}{8} \right\rceil \text{ V}
\]

Note that the output will be inverted!
D/A Converters as Integrated Circuits

- D/A converters are available commercially as integrated circuits.
- Can be classified in three categories.
  - Current output, voltage output, and multiplying type
    - Current output DAC provides the current $I_O$ as output signal
    - Voltage output D/A converts $I_O$ into voltage internally by using an op amp and provides the voltage as output signal
    - In multiplying DAC, the output is product of the input voltage and the reference source $V_{REF}$.
  - Conceptually, all three types are similar
Example

- What will be the analog equivalent of 1001 0001?

\[
I_O = \frac{V_{REF}}{R_{REF}} \left( A_1 \frac{1}{2} + A_2 \frac{1}{4} + A_3 \frac{1}{8} + \cdots + A_n \frac{1}{2^n} \right)
\]

\[
I_O = \frac{V_{REF}}{R_{REF}} \left( DB_7 \frac{1}{2} + DB_6 \frac{1}{4} + DB_5 \frac{1}{8} + \cdots + DB_0 \frac{1}{2^8} \right)
\]

\[
I_O = 5 \text{ mA} \left( \frac{1}{2} + \frac{0}{4} + \frac{0}{8} + \frac{1}{16} + \frac{0}{32} + \frac{0}{64} + \frac{0}{128} + \frac{1}{256} \right) = 2.832 \text{ mA}
\]
How can you generate a sine wave?

- Theoretically the voltages would range from 0 to 5
- How do you change the frequency?

<table>
<thead>
<tr>
<th>Sine D Angle</th>
<th>Pins 9 8 7 6</th>
<th>Vout</th>
</tr>
</thead>
<tbody>
<tr>
<td>270</td>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>258.75, 281.25</td>
<td>1000</td>
<td>0.48</td>
</tr>
<tr>
<td>247.50, 292.50</td>
<td>0100</td>
<td>0.87</td>
</tr>
<tr>
<td>236.25, 303.75</td>
<td>1100</td>
<td>1.21</td>
</tr>
<tr>
<td>225.00, 315.00</td>
<td>0010</td>
<td>1.55</td>
</tr>
<tr>
<td>213.75, 326.25</td>
<td>1010</td>
<td>1.79</td>
</tr>
<tr>
<td>202.50, 337.50</td>
<td>0110</td>
<td>1.99</td>
</tr>
<tr>
<td>191.25, 348.75</td>
<td>1110</td>
<td>2.17</td>
</tr>
<tr>
<td>11.25, 168.75</td>
<td>0001</td>
<td>2.51</td>
</tr>
<tr>
<td>22.50, 157.50</td>
<td>1001</td>
<td>2.63</td>
</tr>
<tr>
<td>33.75, 146.25</td>
<td>0101</td>
<td>2.75</td>
</tr>
<tr>
<td>45.00, 135.00</td>
<td>1101</td>
<td>2.85</td>
</tr>
<tr>
<td>56.25, 123.75</td>
<td>0011</td>
<td>2.96</td>
</tr>
<tr>
<td>67.50, 112.50</td>
<td>1011</td>
<td>3.05</td>
</tr>
<tr>
<td>78.75, 101.25</td>
<td>0111</td>
<td>3.12</td>
</tr>
<tr>
<td>90</td>
<td>1111</td>
<td>3.20</td>
</tr>
</tbody>
</table>
Interfacing with 7-Segment
Interfacing Output Peripherals

- Commonly used output peripherals in embedded systems are:
  - LEDs, seven-segment LEDs, and LCDs; the simplest is LED
- Two ways of connecting LEDs to I/O ports:
  - LED cathodes are grounded and logic 1 from the I/O port turns on the LEDs - The current is supplied by the I/O port called current sourcing.
  - LED anodes are connected to the power supply and logic 0 from the I/O port turns on the LEDs - The current is received by the chip called current sinking.
Seven-segment LEDs

- Often used to display BCD numbers (1 through 9) and a few alphabets.
- A group of eight LEDs physically mounted in the shape of the number eight plus a decimal point.
- Each LED is called a segment and labeled as ‘a’ through ‘g’.
Interfacing Seven-Segment LEDs as an Output (2 of 4)

- Two types of seven-segment LEDs
  - Common anode
  - Common cathode
Interfacing Seven-Segment LEDs as an Output (3 of 4)

- In a common anode seven-segment LED
  - All anodes are connected together to a power supply and cathodes are connected to data lines
- Logic 0 turns on a segment.
- Example: To display digit 1, all segments except b and c should be off.
- Byte 11111001 = F9H will display digit 1.
Interfacing Seven-Segment LEDs as an Output (4 of 4)

- In a common cathode seven-segment LED
  - All cathodes are connected together to ground and the anodes are connected to data lines

- Logic 1 turns on a segment.

- Example: To display digit 1, all segments except b and c should be off.

- Byte 00000110 = 06H will display digit 1.
Interfacing Seven-Segment LEDs to PORTB and PORTC
Seven-Segment Chips

ALPHA/NUMERIC
C/A DISPLAY
Interfacing to Multiple 7-Segments
Using the Simulator

![Diagram of the Simulator Interface]

- **Program Location**: C:\Microchip_projects\Illust9-8 Multiplex Seven Segment.hex
- **Microcontroller**: PIC18F4520
- **Clock Frequency**: 8.0 MHz
- **Last Instruction**: MOVLW 0x02
- **Next Instruction**: MOVWF 0x0110
- **Instructions Counter**: 307
- **Program Counter and Working Register**:
  - PC: 00003E
  - W Register (wREG): 02
- **Special Function Registers (SFRs)**
- **General Purpose Registers (GPRs)**

![7-Segment LED Displays Panel](Diagram of LED Displays Panel)

- **Setup**
- **7-Segment LED Displays Panel**
- **Display Enable**: PORTC, 3
- **Ports**:
  - PORTB: 0, 1, 2, 3
  - PORTB: 4, 5, 6, 7

![LED Color Options](Diagram of LED Color Options)

- **LED Color**
- **Inverted Levels**
- **Inverted Level**

The Simulator provides a comprehensive view of the microcontroller's state and allows for step-by-step execution and visualization of the program's behavior.
Interfacing Input Peripherals

- Commonly used input peripherals in embedded systems are: DIP switches, push-button keys, keyboards, and A/D converters.
- DIP switch: One side of the switch is tied high (to a power supply through a resistor called a pull-up resistor), and the other side is grounded. The logic level changes when the position is switched.
- Push-button key: The connection is the same as in the DIP switch except that contact is momentary.
Interfacing Dip Switches and Interfacing LEDs
Liquid-Crystal Display (LCD)
Read/Write (RW) connects to the ground if processor does not read data from LCD.

LED+ and LED- pins provide 5V voltage for backlights.

Enable (E) pin provides clock signal.

Register Select (RS)
- RS = 0: data on the data bus is a command
- RS = 1: data on the data bus is actual data.
Saves four pins for the processor
For each 8-bit data, transfer the upper 4 bits first and then the lower 4 bits.
HD44780 Controller: Internal Diagram

- **DDRAM Address Register**: 0x00
- **Data Register**: 0x52
- **Command Register**: 0x00
- **Data**: 0
- **Register Select (RS)**: 0
- **ROM for user-defined fonts**: 1110
- **ROM for standard fonts**: 1000
- **Character Generator (CG)**: 0000
- **Display Memory (DDRAM)**: (It stores all characters received.)
- **Binary Encoding**: 1110, 1001, 1001, 1110, 1010, 10010, 10001, 00000
- **Image Pattern**:
  - 1 1 1 1 0
  - 1 0 0 0 1
  - 1 0 0 0 1
  - 1 1 1 1 0
  - 1 0 1 0 0
  - 1 0 0 1 0
  - 1 0 0 0 1
  - 0 0 0 0 0
- **LCD Display**: R
- **COM Signals**: LCD Display
- **SEG Signals**: LCD Display

**Legend**:
- **“R” ASCII 0x52**: Data Register
- **DB**: DDRAM Address Register
- **addr**: Address
- **data**: Data
- **Table Lookup**: Display Memory (DDRAM)
- **RAM for user-defined fonts**: ROM for user-defined fonts
- **ROM (Read-only Memory) for standard fonts**: ROM for standard fonts
- **Character Generator (CG)**: Character Generator (CG)
Clock Signal

- On the falling edge of E (Enable), data are latched into the data or command register
  - RS = 0: command; RS = 1: data
- E is a data transmission clock signal provided by the processor
- A nibble = 4 bits
- For each byte, the upper nibble is sent out first.
void LCD_Pulse(void) {
    LCD_Port->ODR |= 1<<LCD_EN; // Set E high
    Delay(4); // Delay 40us
    LCD_Port->ODR &= ~(1<<LCD_EN); // Set E low
    Delay(4); // Delay 40us
}

8-bit Data Bus Mode

4-bit Data Bus Mode

Latch 8 bits

Latch upper 4 bits

Latch lower 4 bits
Example Signals

- Sending “hello”
### Sending Command and Data

<table>
<thead>
<tr>
<th>Sending Command to LCD</th>
<th>Sending Data to LCD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void LCD_SendCmd(uint8_t c) {}</strong></td>
<td><strong>void LCD_SendData(uint8_t c) {}</strong></td>
</tr>
<tr>
<td>// RS: 0 = command, 1 = data</td>
<td>// RS defaults to 1</td>
</tr>
<tr>
<td>LCD_Port-&gt;ODR &amp;= ~(1&lt;&lt;LCD_RS);</td>
<td>// Send Upper 4 bits</td>
</tr>
<tr>
<td>// Send Upper 4 bits</td>
<td>LCD_PutNibble( c &gt;&gt; 4 );</td>
</tr>
<tr>
<td>LCD_PutNibble( c &gt;&gt; 4 );</td>
<td>LCD_Pulse();</td>
</tr>
<tr>
<td>LCD_Pulse();</td>
<td>// Send Lower 4 bits</td>
</tr>
<tr>
<td>// Send Lower 4 bits</td>
<td>LCD_PutNibble( c &amp; 0xF );</td>
</tr>
<tr>
<td>LCD_PutNibble( c &amp; 0xF );</td>
<td>LCD_Pulse();</td>
</tr>
<tr>
<td>// Return to default</td>
<td></td>
</tr>
<tr>
<td>LCD_Port-&gt;ODR</td>
<td>= 1&lt;&lt;LCD_RS;</td>
</tr>
</tbody>
</table>
| }                                                           | }
## Command Format

<table>
<thead>
<tr>
<th>RS</th>
<th>RW</th>
<th>DB7</th>
<th>DB6</th>
<th>DB5</th>
<th>DB4</th>
<th>DB3</th>
<th>DB2</th>
<th>DB1</th>
<th>DB0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Clear screen and set DDRAM address to 0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Return home (set DDRAM address to 0)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>I/D S Entry mode set</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Control on/off of display, cursor, and blink</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>SC</td>
<td>RL</td>
<td>Cursor/display shift</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>DL</td>
<td>N</td>
<td>F</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A5</td>
<td>A4</td>
<td>A3</td>
<td>A2</td>
<td>A1</td>
<td>A0</td>
<td>Set code generator CG-RAM address A5-A0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A6</td>
<td>A5</td>
<td>A4</td>
<td>A3</td>
<td>A2</td>
<td>A1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>BF</td>
<td>A6</td>
<td>A5</td>
<td>A4</td>
<td>A3</td>
<td>A2</td>
<td>A1</td>
<td>A0</td>
<td>Ready busy flag (BF) and address A6-A0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>D7</td>
<td>D6</td>
<td>D5</td>
<td>D4</td>
<td>D3</td>
<td>D2</td>
<td>D1</td>
<td>D0</td>
<td>Write data D7-D0 to DDRAM or CG-RAM</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>D7</td>
<td>D6</td>
<td>D5</td>
<td>D4</td>
<td>D3</td>
<td>D2</td>
<td>D1</td>
<td>D0</td>
<td>Read data D7-D0 from DDRAM or CG-RAM</td>
</tr>
</tbody>
</table>
LCD Initialization

1. Wait for ≥ 30 ms
2. Send command 0x30
3. Wait for ≥ 4.1 ms
4. Send command 0x30
5. Wait for ≥ 160 μs
6. Send command 0x30
7. Wait for ≥ 160 μs

8-bit data bus mode

8a. Send command 0b0011NF**
8b. Wait for ≥ 40 μs

Yes

Does data bus have 8 bits?

No

4-bit data bus mode

8a. Send command 0x2 to switch to 4-bit mode
8b. Wait for ≥ 40 μs
8c. Send command 0b0010NF**
8d. Wait for ≥ 40 μs

9. Send command 0x01 to clear display
10. Wait for ≥ 1.5 ms
11. Send command 0x06 to set entry mode
12. Wait for ≥ 40 μs
13. Send command 0x0F to control display ON/OFF
14. Wait for ≥ 40 μs

Initialization completes
Customized Font

// Select CGRAM and set address to 0x00
LCD_SendCmd(0x40 + 0x00);
Delay(4); // Wait > 39us

// Define smile face
LCD_SendData(0x00); // 1st row byte
LCD_SendData(0x0A); // 2nd row byte
LCD_SendData(0x0A); // 3rd row byte
LCD_SendData(0x0A); // 4th row byte
LCD_SendData(0x00); // 5th row byte
LCD_SendData(0x11); // 6th row byte
LCD_SendData(0x0E); // 7th row byte
LCD_SendData(0x00); // 8th row byte

// Select display RAM & set address to 0
LCD_SendCmd(0x80); // First character
Delay(4); // Wait > 39us
LCD_SendData(0x00); // Display the font