GUIDE TO HARDWARE I

Required Study Material

Prepared by Engrs. Ashraf Al-Suyyagh – Enas Ja'ra

We present this guide to serve as a reference for many electrical and electronic components which students are to use throughout the lab course and the final project. We attempt to cover all required material such that all students will be on the same level of basic knowledge. The components are presented in the context of their use in embedded systems where they are interfaced with microcontroller devices.

1. Clocking sources

Microcontrollers need clocking sources to synchronize their functions. An ideal clock is a square wave function. There are two components which provide such capability: oscillators and resonators. Basically the faster the clocking source; the faster the processing speed. However, fast processing also requires more current and therefore generates more heat. PIC 16F series can operate up to 20MHz whereas the 18F series has an operation speed of up to 40MHz. Details of operation principles, interfacing and calculations are presented in these following subsections

1.1. Oscillators

Oscillators usually built from crystal (most notably Quartz crystal– as in wristwatches) have simple operation principle: *The use of the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency*. To make things clear: we will start to define terms and simplify operation principles; a piezoelectric material is one which has the characteristic of changing shape when voltage or an electric field is applied to it then reverts back to its original shape once the induced voltage/electric field is removed. While switching back to its original state, the material itself generates an electric field and thus a voltage with very precise frequency which is the oscillator frequency. So simply, to get the oscillator to work, feed it with voltage at one pin and use the output frequency from another. In this manner, the oscillator can be modeled as an RLC circuit with a specific resonance frequency as you should have already learnt in the Circuits II course. Yet, since crystals are **mechanical** devices which vibrate at their resonance frequency, they are not that precise, that is, they don't produce an ideal square wave function of which the high level of the signal is fixed at 50% of the period's time, and instead it can be any time in between 40% - 60% of the period. Therefore, to account for such errors, the clock signal is divided by a certain fixed value to minimize error effects, in the PIC MCUs, the input clocking source is divided by four. In that manner, every four pulses of the original signal will generate one pulse in the new signal.

Figure 1 below further clarifies the idea; the top signal is an ideal square wave signal, the duty cycle of which is exactly half the period. Notice how the middle signal – the actual signal as generated from a clocking source – deviates from the ideal, the duty cycle differs (red). By dividing the cycle by a fixed value – bottom signal – we minimize the errors for the new signal hides the frequent changes in between successive pulses.

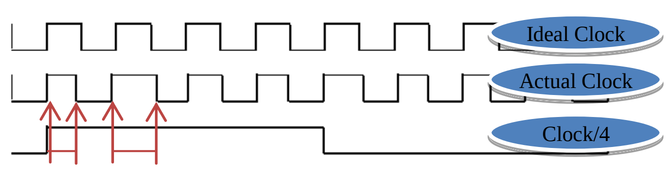

Figure 1 - Clock

Oscillators come in variety of form factors and speeds. Figure 2 shows two common form factors of oscillators, the 2-pin oscillator (left) and 4-pin oscillator (right). You can place the 2-pin oscillator in either direction on the OSC pins of the microcontroller, the 4-pin oscillator only uses three pins and the forth is not connected, the pins are GND, Vcc and output. Refer to the datasheet of the oscillator to determine which pin is which.

Figure 2 - Two Common Shapes of Oscillators

After connecting an oscillator, one should explicitly specify to the PIC which oscillator speed range and type it should expect. This option can be set in either the MPLAB configuration bits window prior to programming or by explicitly specifying it in the configuration word in the source code. There are four options:

- **XT** – Crystal: 1-4 MHz
- **HS** – High Speed: >= 4 MHz, and with ceramic resonators.
- **LP** – Low Power: <= 200 KHz,
- **RC** – Resistor-Capacitor (if you build the resonance circuit by yourself)

Along with the crystal, two capacitors of approximately (10-33) pF are required, crystal needs loading capacitors to work at the exact operating frequency (i.e. to get a stable oscillation from the crystal oscillator) and for noise immunity. An important note though is that the operating frequency is not fixed and that it varies with temperature. The advertised frequency is usually specified at room temperature 25°C, clocks slow down when temperatures increase or decrease from the nominal room temperature. For accurate timing one needs to know the operating frequency at different temperatures, for this we can use the following formula (assume all other parameters are at their recommended values):

$$F = F_0(1 - P\ x\ (T-T_0)^2)$$

Where:

- **T** is the expected temperature in Celsius
- $\mathbf{T_0}$ room temperature 25°C
- $\mathbf{F_0}$ advertised oscillator frequency
- **F** actual frequency at temp T
- **P** is the frequency stability coefficient (obtained from datasheet – units in ppm)

Example:

A 32 kHz oscillator with a frequency stability coefficient of 0.004 ppm running in arid environment where average temperature is 35°C will actually have an oscillation frequency of:

$32000(1 - 0.004\ (35\text{-}25)^2) = 19.2\text{kHz}!!$

From the above example, we clearly show the importance of considering temperature effects upon the frequency of operation.

Figure 3 shows a typical crystal interfacing to a PIC

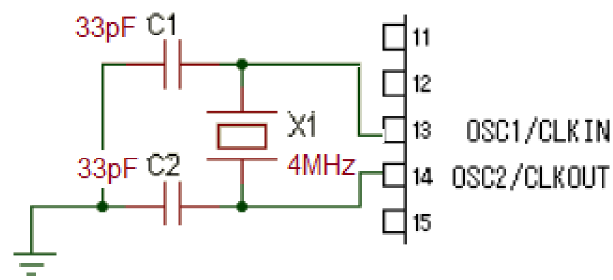

Figure 3 - Interfacing a 4MHz Crystal to PIC16F877A

1.2. Resonators

Resonators are made of high-stability piezoelectric ceramics and share the same operating principles of oscillators but differs in that it consists of a voltage-variable capacitor that acts in some ways like a quartz crystal. The thickness of the ceramic substrate determines the resonance frequency of the device. Resonators have either two or three pins. They need not loading capacitors. They have a similar connection as the oscillator in Figure 3, if a three lead resonator is used the middle pin is connected to GND. Figure 4 shows a typical resonator.

Figure 4 - A Typical Resonator

2. REGULATORS

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level at the output given varying voltage at the input. Depending on the part number and manufacturer specifications, regulators can take a limited range of input voltages and produce a limited range as well. Regulators most often have metallic heat sinks attached to dissipate heat more efficiently. Many commercial regulators regulate fixed voltages, commonly 3, 5, 9, 12 and 15 volts. One must be cautious to the input and output currents to and from the regulator, too much input current than specified will overheat and eventually burn the device. Too much load current will have the effect of regulator output voltage drop!

There are two main series of regulators: the 78xx and 79xx series. The 78 represents a family of regulators which regulates positive voltages and the 79 family regulates negative ones. The xx part is the output voltage of the device.

Examples:

- 7805: 5V DC Regulator
- 7905: -5V DC Regulator
- 7808: 8V DC Regulator
- 7909: -9V DC Regulator

Regulators have three pins, one connected to the input voltage source, another to the circuit and the middle one is shared ground in between the input source and the circuit. Figure 5 shows a typical regulator.

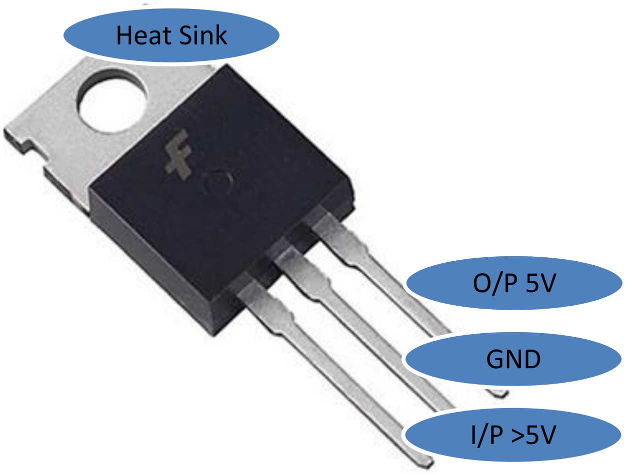

Figure 5 - A Typical 7805 Regulator

2.1. More on Regulator Heat Sink

The heat sink is a component designed to lower the temperature of an electronic device by dissipating heat into the surrounding air. As a general rule the input voltage should be limited to 2 to 3 volts above the output voltage. The LM78XX series can handle up to 36 volts input, be advised that the power difference between the input and output appears as heat. So heat which will be dissipated by the chip during the voltage regulation process. This can cause the chip to heat up, and so a heat sink is often used to speed up heat removal and prevent overheating.

Figure 6 shows a typical in-circuit connection for the 7805 regulator. A couple of coupling capacitors (between 10 uF and 47 uF) are required on the input (V-IN) and output (V-OUT) and connected to ground. Coupling capacitors are used for good regulation and to reduce unwanted AC signals riding on DC supply circuits (Noise)

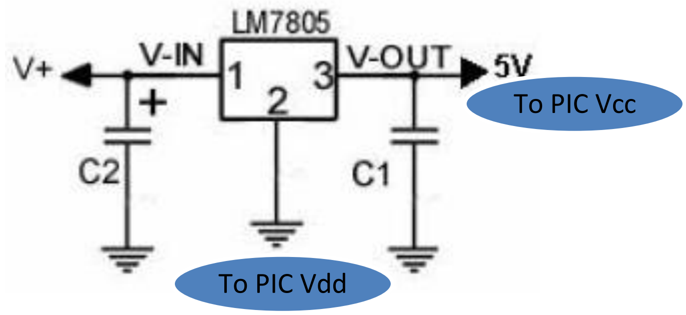

Figure 6 - 7805 Circuit Diagram

3. PIC RESET CIRCUIT

As you should have already learnt in the course, PIC MCUs already have a master clear pin called MCLR, keep in mind that this is an active low pin. Therefore, PIC reset circuitry is simply constructed by wiring a switch to MCLR, and when pushed gives logical ‘0’ or GND to this pin. This has an effect of resetting the microcontroller, clearing all RAM and starting program execution from the beginning. A **pull up resistor** circuitry is used to hold the input at logic “1” state as long as the reset button is not pressed. Figure 7 shows the circuit diagram of the reset circuitry.

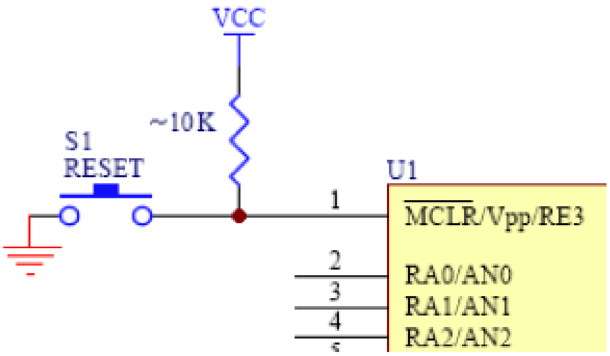

Figure 7 - PIC Reset Circuit

4. Pull-Up and Pull-Down Resistors

Pull-up resistors are used in electronic logic circuits to ensure that inputs to logic systems settle at expected logic levels if external devices are disconnected. The idea of a pull-up resistor is that it weakly "pulls" the voltage of the wire it's connected to towards 5V (or whatever voltage represents logic "high"). However, the resistor is intentionally weak (high-resistance) enough that, if something else strongly pulls the wire toward 0V, the wire will go to 0V. Pull-down resistors operate in a similar fashion where they are initially pulled down to logic 0 through a connection to ground, but when a source pulls it up toward logic high it will change state. Pull up and pull down resistors are used with switches and push buttons to fix the state of the pin connected to the switch at a predetermined state and not be kept floating. Pull up and pull down resistors take a minimum value of 4.7kΩ.

5. Light Emitting Diodes

Light emitting diodes or LEDs are semiconductor light sources used as indicator lamps in many electronic devices. Modern versions are available across the visible, ultraviolet and infrared wavelengths, with very high brightness and come in a variety of shapes and sizes. The physics behind LED operation is covered in the Electronics I course and will not be offered here. Figure 8 shows the different color spectrum of LEDs.

Figure 8 - Different Colors of LEDs

In order to switch a LED on, forward current must pass from the anode to the cathode, but how to determine which pin of the LED is anode and which is cathode, generally, there are two ways:

1. The longer lead is anode, the shorter is cathode.
2. The cathode has a flat surface as shown in Figure 9

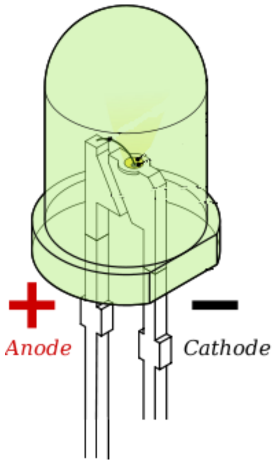

Figure 9 - Determining Cathode and Anode in LED

Resistors with values in between 220Ω to 1kΩ are placed in between the voltage source (often 5 to 9V or even more) and the anode to limit the current entering the LED or else it will burn. The lesser the resistor value, the brighter the LED shines (Ohms Law). In this case these resistors are called current limiting resistors.

Figure 10 shows how to interface a LED to the PORTC pin 1

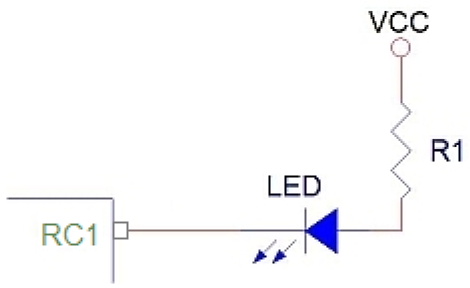

Figure 10 - LED Interfacing

Not only are LEDs used as discrete components but are also the building blocks of LED Matrices and 7-Segment displays. Figure 11 shows an example of a LED matrix.

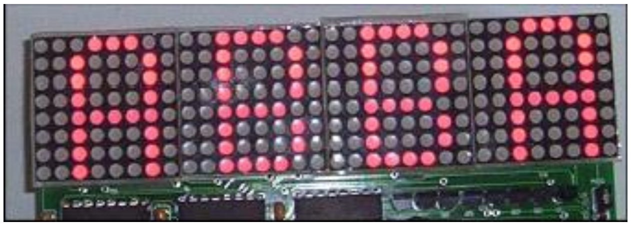

Figure 11 - LED Matrix

6. SEVEN-SEGMENT DISPLAYS

A Seven-Segment display, as its name implies, is composed of seven segments (or technically of seven LEDs) which can be individually switched on or off. This ability to individually control each segment and the layout in which these segments are distributed allows for the representation of the numerals and some characters. If the anode ends of all LEDs are connected together, it is called common anode display. If the cathodes of all LEDs are connected together, it is called a common cathode display. To switch a LED on in a common cathode configuration, you have to send logic high to the segment pin. Conversely, to switch a LED on in a common anode configuration, you have to send logic low to the segment pin.

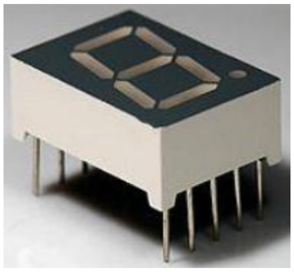

Figure 12 - A Single Unit 7-Segment Display

Seven-Segment displays can be purchased in single units encompassing one, two, three or even four displays in the same package. Digit Multiplexing techniques are widely used to allow for the multiple displays to share the same segment pins simultaneously while each displaying a different numeral or character. Figure 12 shows a typical single unit seven segment display while Figure 13 shows the typical layout of segment pins for the common cathode and common anode configurations.

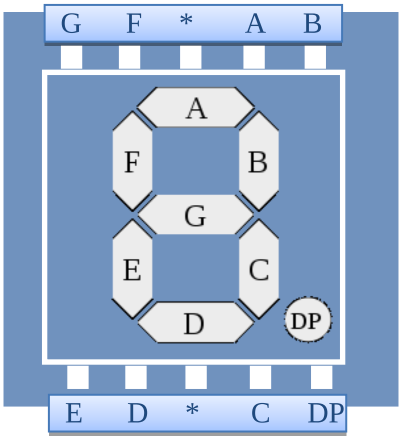

Figure 13 - Seven-Segment Display
*Means Vcc for common Anode, and GND for common Cathode

Interfacing a Seven-Segment display is independent of the type of the module, whether it is common anode or common cathode, only the logic level sent to the display differs. Finally, since the display is basically LEDs, current limiting resistors are used for each segment.

7. SWITCHES AND PUSH BUTTONS

Switches and pushbuttons have similar operation, to switch the input level between two alternating levels, the only difference is that a push buttons only retains the level as long as it is pressed and reverts back to its prior state once the press effect is gone.

7.1. Switches

A switch is an electrical component which can break an electrical circuit, interrupting the current or diverting it from one conductor to another. There are four types of switches:

- SPST Single Pole, Single Throw: SPST is simple on-off switch. This type is simply used for turning something on and off
- SPDT Single pole, double throw: SPDT switches are useful if you want to supply some instrument with two different voltages or divert current between two different paths.
- DPST Double pole, single throw: A Double-pole Single-throw switch is simply two SPST switches together. It allows you to switch two separate circuits on and off at once.
- DPDT Double pole, double throw: DPDT switches have six terminals and allow one to switch poles between two different circuits.

Figure 14 below shows typical SPDT switch.

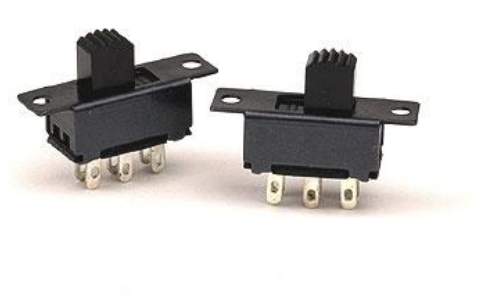

Figure 14 - SPDT Switches

Switches use pull-up or pull down resistors to hold the input voltage supplied to the PIC at a predetermined level, only when the switch is used does the voltage level change. Figure 15 shows the circuit diagram of interfacing an SPST switch to PIC. Note that the pull up or pull down resistors take a minimum of 4.7kΩ, you can determine the exact value using Ohm's Law.

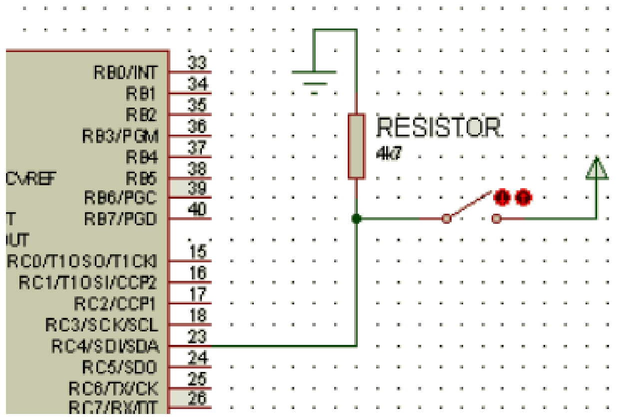

Figure 15 - Interfacing an SPST Switch – Same circuit is used to interface a push button.

7.2. Push Buttons

There are two types of push buttons:

- Normally closed push button (abbreviated NC) is one that normally gives logic one and when pressed gives logic zero.
- Normally open push button (abbreviated NO) is one that normally gives logic zero and when pressed gives logic one.

Push buttons are interfaced in the exact same way as an SPST switch shown in Figure 15. Figure 16 shows a push button.

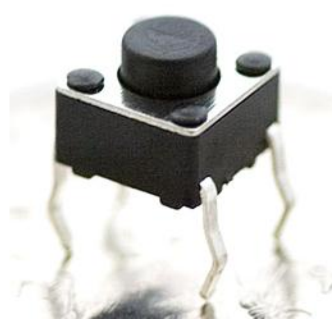

Figure 16 - A Push Button

7.3. Mechanical Switch De-bouncing

Push-buttons and switches are often used to provide input to digital systems. However, mechanical switches do not open or close cleanly. When a switch is pressed it makes and breaks contacts several times before settling into its final position. This causes several transitions or "bounces" to occur. To correct this situation a **de-bounce** circuit is connected to the switches, thus removing the series of pulses generated by the mechanical action of the switch. Figure 17 shows a circuit which suffers from bouncing effects.

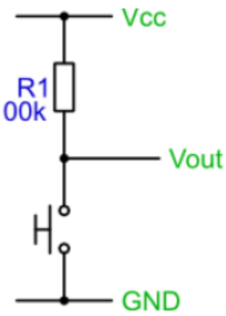

Figure 17 - A Circuit Suffering from Mechanical Bouncing Problem

Figure 18 shows an oscilloscope captured image clearly showing the bouncing effect.

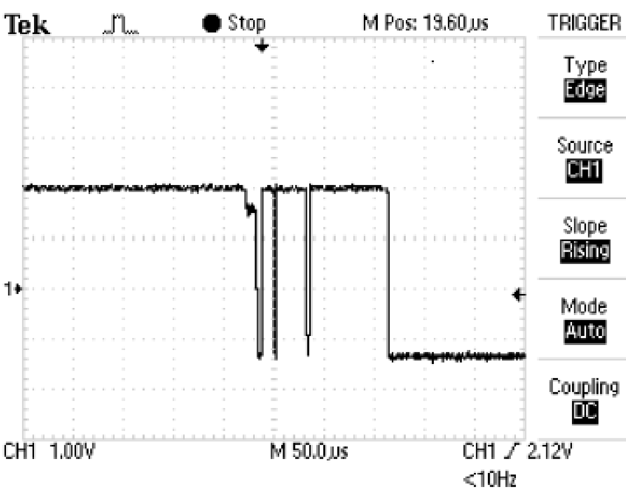

Figure 18 - Bouncing Effect. Note that it approximately lasts for 150 us

Solutions

There are two solutions to the bouncing problem: hardware and software approaches.

Hardware de-bouncing: The most basic circuit used to de-bounce a switch is shown in Figure 19. It consists of a resistor and a capacitor in series. The resistor and capacitor values must be chosen such that the RC time constant is greater than the bounce time. The output is then connected to a Schmitt trigger. At the start of operation the capacitor is charged to Vcc and the output is at 5 volts, when the switch is closed, the capacitor starts discharging smoothly and this filters out the bounces. The Schmitt trigger is a comparator which gives a high output if the input value is over a certain threshold, and a low output if below. In this case, the Schmitt trigger is necessary because the smoothed out value from the capacitor is neither high nor low but an exponential signal which digital devices don't understand, therefore it is up to the Schmitt trigger to convert it to logic highs and lows.

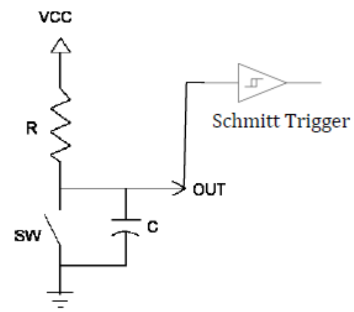

Figure 19 - A Hardware De-Bouncing Solution

Software de-bouncing: The basic idea is to read the switch input signal after some time interval guaranteed to be larger than the duration which the bouncing lasts and thus skip any short-lived bounces. In. Figure 18, one can read the signal after 200 µs.

Which is better: Software or Hardware de-bouncing?

It depends on your application needs; if time is critical and speed is important, you need not waste cycles in generating delays and therefore hardware solutions are preferable. If, however, you are developing a simple small-scale project where you want to reduce the hardware costs, then the software approach is better. All in all, you need to compromise and choose depending on your application and development needs