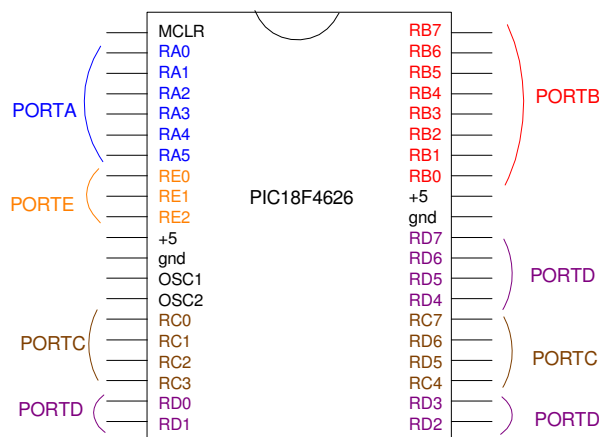


Binary Inputs & Counters



The PIC18f4620 chip has 33 I/O lines split into five ports:

	PORTA	PORTB	PORTC	PORTD	PORTE
Pins	2..7	33..40	15..18, 24..26	19..22, 27..30	3
Binary Input	5	8	8	8	3
Binary Output	5	8	8	8	3
Analog Input	5	5	-	-	3

Binary Input (Hardware)

PORTA, PORTB, and PORTC can read TTL level inputs (0V = 0, 5V = 1). If you can convert whatever you're measuring to a 0V / 5V signal, you can read it with a PIC. This section gives several examples.

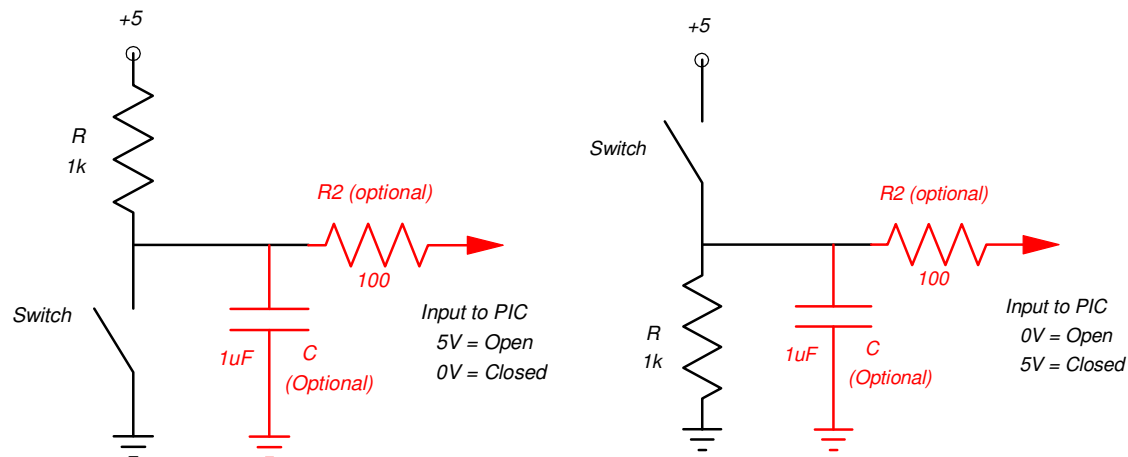
Reading a SPST (momentary) switch:

Switches come in several varieties

- SP: Single Pole. There is a set of electrical connections
- DP: Double Pole. There are two sets of electrical connections
- ST: Single Throw. Two leads are open or closed.
- DT: Double Throw. A center lead can connect to the left or right lead.

A momentary switch is a SPST: it has a single set of leads that are connected or not.

One circuit which can convert a SPST switch to 0V / 5V follows:



On the left circuit,

- When the switch is open, the lead to the PIC is pulled high to +5V. The PIC reads logic 1.
- When the switch is closed, the lead to the PIC is grounded. The PIC reads logic 0.

On the right circuit, logics 1 and 0 are switched.

The optional components have the following function:

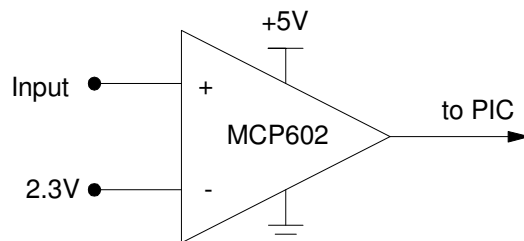
- **C:** Mechanical switches often 'bounce' when closed. In some situations, this 'bounce' can be read by the processor as several openings and closings rather than just one. To fix this problem (termed debouncing), a filter can be used, either in hardware (adding a capacitor) or software (look for a '1' or '0' which lasts so long.) **C** is chosen so that the settling time is short enough for events you're trying to measure ($T_s = 4/RC$).
- **R2** is a buffer to protect the PIC processor. The pin on the PIC processor that the switch is connected to *should* be set up as an input. Inputs have high impedance, so the current in **R2** should be zero and **R2** has no effect on the circuit's operation. If the PIC pin is accidentally set up as an output pin, however, **R2** acts as a buffer preventing you from tying an output pin to 0V or +5V directly.

Binary Input & Voltages

Design a circuit which outputs

- +5V when the input is more than 2.3V
- 0V when the input is less than 2.3V

Solution: Use an op-amp (such as the MCP602) in your lab kit. Connect the + power to +5V and the - power to ground. The output of the op-amp is limited to the power supply, 0V and +5V.



Binary Inputs & Temperature Without Hysteresis: Comparator

Design a circuit which outputs

- +5V when the temperature is above +20C
- 0V when the temperature is below +20C

Solution: To measure temperature, use the thermistor in your lab kit. This thermistor has a resistance vs. temperature relationship of (approximately)

$$R = 1000 \cdot \exp\left(\frac{3930}{T} - \frac{3930}{298}\right) \Omega$$

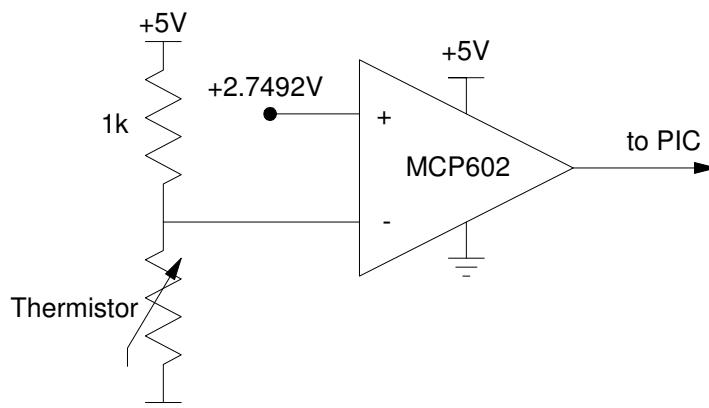
where T is the temperature in degrees Kelvin.

$$\text{Kelvin} = \text{Celcius} + 273$$

At +20C (293K), it should have a resistance of

$$R(20C) = 1250\Omega$$

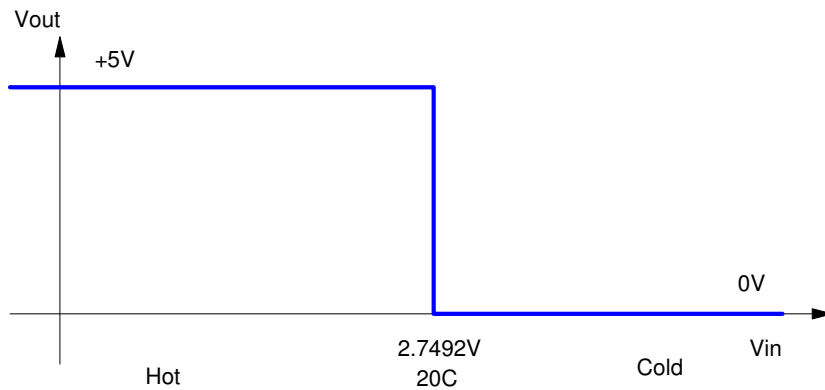
To convert resistance to voltage, use a voltage divider. Assume a 1k resistor.



At 20C, the voltage is 2.778V. As temperature goes up, the resistance drops and so does the voltage. So, connect the thermistor to the - input of the op-amp. Connect +2.778V to the + input.

The input/output voltage characteristic of this circuit is then as follows:

- At 20C (2.778V), the output switches.
- For voltages below 2.778V ($T > 20C$), the output goes to 5V
- For voltages above 2.778V ($T < 20C$), the output goes to 0V



Input / Output Characteristic of the Comparator

Binary Inputs & Temperature With Hysteresis: Schmitt Trigger

Design a circuit with hysteresis:

- The output switches to +5V when the temperature goes above +20C
- The output switches to 0V when the temperature drops below +15C

Solution: Use the above circuit for a start. Add positive feedback to the op-amp to create a Schmitt trigger.

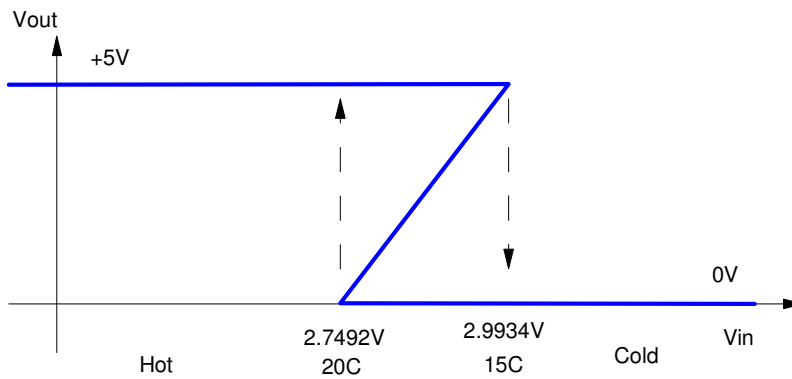
At +20C,

- $R = 1250$ Ohms
- $V = 2.778V$.

At +15C

- $R = 1575$ Ohms
- $V = 3.059V$

The input / output you want in terms of voltage is then as follows



Input - Output Characteristic With Hysteresis. The output goes low for $T < 15C$ ($V > 2.99V$) and high for $T > 20C$ ($V < 2.74V$)

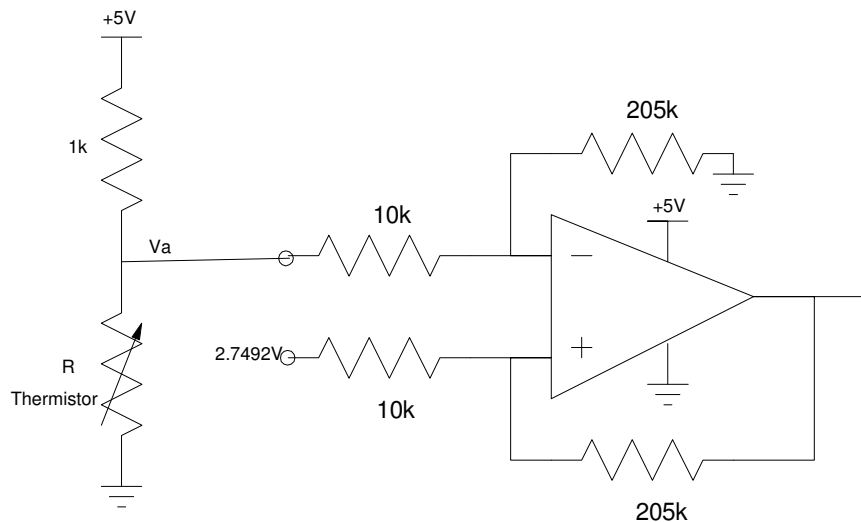
The voltage difference is 0.2443 volts. The feedback reduces the +5V swing at the output to a +0.2443 volt swing at the + input. Pick the feedback resistors for a gain of

$$gain = \left(\frac{\text{change in output}}{\text{change in input}} \right) = \left(\frac{5V}{0.2443V} \right) = 20.5$$

A 10k and 205k resistor works.

When the input voltage is large (temperature is cold), the output should be small. Connect to the - input.

When the output is 0V, you are on the verge of switching when $V_{in} = 2.7492V$. Make the offset voltage 2.7492V.



Circuit with Hysteresis (Schmitt Trigger). Output goes high for $T > 20C$, low for $T < 15C$.

Note that these circuits also work with light, pressure, motor speed, etc. All you need is a sensor which outputs voltage (for the first example) or resistance (for the second or third example).

- CdS sensors convert light to resistance
- Photovoltaic sensors convert light to voltage (current actually...)
- Gas sensors convert O₂, CO₂, methane, etc to a resistance or voltage
- Strain gages convert strain (or weight or pressure) to resistance
- Tachometers convert motor speed to a voltage.

Counters

Once you have whatever you're measuring converted to TTL levels (0V / 5V), you can write a program to do things, like count. For example, turn your PIC into an up/down counter

Example: Hungry-Hungry Hippo Game

Input:

- Push buttons RB0 and RB7

Output:

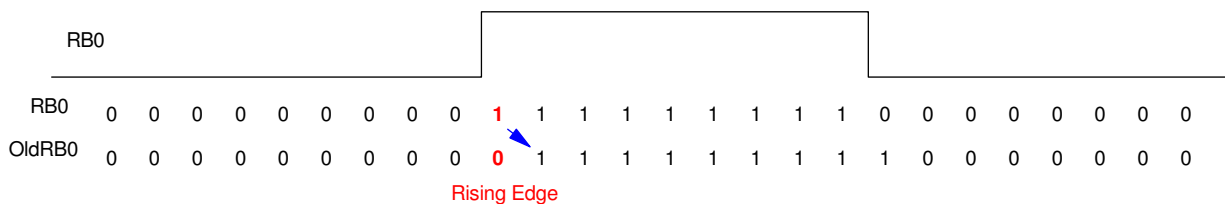
- PORTC and PORTD

Relationship:

- Start with PORTC = PORTD = 0
- Each time you detect a rising edge on RB0, increment PORTC by one
- Each time you detect a rising edge on RB7, increment PORTD by one

Note: Detect a rising edge on RB0 when

- RB0 = 1 and
- OldRB0 = 0



Flow Chart & Assembler Code

```

; Hungry Hungry Hippo
; Count how many time you press a button
; RB0: Display on PORTD
; RB7: Display on PORTC

```

```

#include <pl8f4620.inc>

```

```

A equ 0
B equ 1

```

```

org 0x800

```

```

clrf TRISA
movlw 0xFF
movwf TRISB
clrf TRISC
clrf TRISD
clrf TRISE
movlw 0x0F
movwf ADCON1

```

```

movff PORTB, A
movff PORTB, B

```

```

L1:
btfss A, 0
goto L2
btfsc B, 0
goto L2
incf PORTD, F

```

```

L2:
btfss A, 7
goto L3
btfsc B, 7
goto L3
incf PORTC, F

```

```

L3:
movff A, B
movff PORTB, A
goto L1

```

```

end

```

