PIC Control of the Mini-Sumo Robot

Mini-Sumo Robotics Project

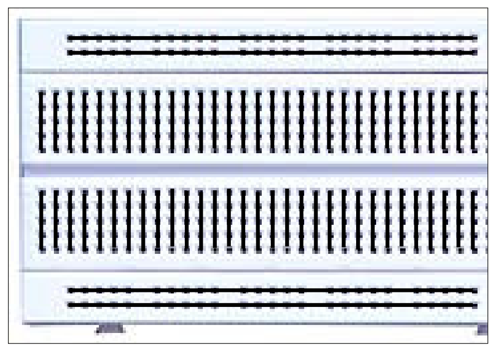
Worksheet #8

Introduction

By now you have hopefully got your mini-sumo robot built to the point where the body is built, the gearbox and wheels attached, and you are ready to demonstrate that you can use a PIC to make your robot move and turn. To do this you will need to attach a number of components to your breadboard.

The Breadboard

The breadboard is a useful device for prototyping electronic circuits. It allows you to connect different components together in different ways, and to try new ideas until you get them right. Typically, once you have demonstrated that your circuit works on a breadboard, you will use that knowledge to produce a printed circuit board, and solder your components in place. This results in a smaller, more reliable, but also (relatively) permanent arrangement of parts.



The breadboard at right indicates how the holes in a breadboard are connected. Each vertical line shows that the five holes underneath it are connected. Each horizontal line shows that the holes underneath it are connected. Typically, the long, horizontal, "header" lines are used to provide a power supply line and a ground line on each side of the board. The horizontal gap down the middle of the board is sized just right to allow IC's (integrated circuits, or "chips") like the PIC to be placed horizontally, with one row of pins on one side of the gap, and one row of pins on the other. To connect components you use fine gauge (20-24 ga) wires as "jumpers", or simply use the "legs" of the components.

Due to the size restrictions placed on mini-sumo robots, typical 830 hole breadboards will not fit. They do, however, work very nicely when cut in half using a bandsaw or hacksaw. If you cut straight, you should end up with a board similar to that shown above.

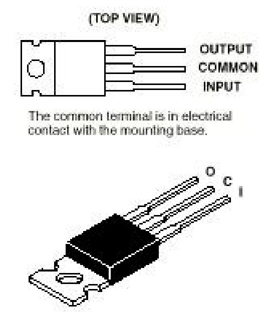
The Components

The main component of our circuit, of course, is the PIC chip. Typically we have used the PIC16F84 or PIC16F84A chip, however newer, more powerful – and less expensive PICs are available. The PIC16F627 and 16F628 appear identical to the older F84, except for the labelling on the back of the chip*. Fortunately, they also have very similar "pin-

* There are other, minor differences, related to programming the newer PICs. Information on upgrading is freely available on many web sites, and the new chip's abilities (programmable voltage reference, on-chip comparators, etc.) are well documented in the datasheet available from www.microchip.com

outs" to the older chip, so these instructions will work for all four different models of PIC. For more information on the PIC, it's I/O Ports and function of its pins, refer to previous worksheets.

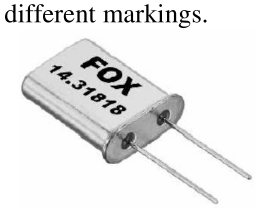
In order to power the PIC we need a stable voltage supply of about 5 volts. To provide a precise 5v supply, we will use an LM7805 voltage regulator, as shown here*. Although the LM7805 looks like a simple transistor, with just three "legs", it is actually an integrated circuit that can take almost any input voltage above 5 volts and "regulate" it down to five volts. As this wastes energy, especially when large voltage drops and large currents are involved, the LM7805 even contains temperature monitoring hardware that will shut it down before it overheats. The hole at the top of it, and metal back are designed for connecting to a heat sink if the need arises. Since in our circuit the 7805 will be providing current for just the PIC, not the motors, heat should not be an issue.



The LM78xx voltage regulator and pin-out.

Another component needed to support the PIC16F84 and F84a is a quartz crystal. The crystal sets the "clock speed" for the PIC, and allows it to keep time very accurately. The crystal we typically use is the Fox040a crystal resonator, which vibrates at 4 Mhz. The crystal is connected to ground by 2 33pF capacitors, which help it to start oscillating. Other crystals, including some with clock speeds up to 20Mhz will also work with the right PIC chip, as will a variety of other types of resonator, including a simple RC (resistor-capacitor) oscillator. Interestingly, the PIC16F627 and 628 chips do not need an external crystal for most applications as they have a built-in 4Mhz oscillator circuit, which frees up the two "clock" pins for other I/O duties. If you are using the PIC16F62x for tasks where precise timing is critical (such as serial communications), however, you may wish to use a crystal to get the very precise clock needed for these applications.

A 14Mhz quartz crystal The 4Mhz one has different markings.



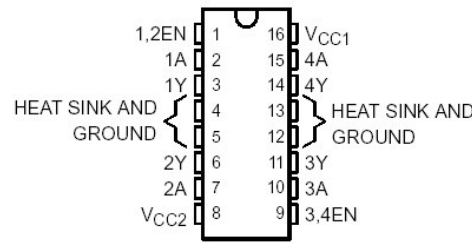
If the PIC is the "brains" of the circuit, the "muscle" for the mini-sumo robot comes from an L293D "quadruple half-H motor driver" chip. While that sounds complex, an "H-bridge" is simply a circuit that allows you to change the polarity of the voltage to a DC motor, thus changing the direction of rotation of the motor. As this chip is a "quadruple half H-bridge" it means we have two H-bridge circuits, and can thus control two DC motors – which is precisely what we need to do. The L293D can handle up to about 1 amp per motor, which is well suited for the little motors on the Tamiya gear box,

* The PIC will actually run just fine with anything from about 3.5-5.5 volts, which means that the 4xAA battery pack should be able to power the PIC as well as the motors. Unfortunately, we discovered that the internal resistance of the (alkaline) battery pack was high enough that when both motors were turned on, the voltage across the battery pack would drop below 3 volts, causing the PIC to reset. Isolating the PIC power supply from the motor power supply as we do in this design (using a separate 9V battery with 5V regulator to power the PIC) has been a reliable solution. Users seeking a more elegant solution may wish to experiment with Ni-Cd, NiMH or Lithium based batteries (or different sized cells) which may provide lower internal resistance, or higher voltage thus eliminating the 9V battery and regulator from the design.

however the chip may heat up with prolonged use. If the chip gets too hot, internal circuitry will shut it down until it cools. Other chips that may be used are the 754410 (a pin-compatible upgrade for the L293D), or the L298, which although it can handle more current (3 amps) may not fit well into the breadboard.

The connection diagram for the L293D is shown at right, along with the "tri-state" logic for each of its outputs. Although we are not going to be using the tri-state abilities of the L293D, it is worth mentioning what tri-state logic is, as the PIC is also a "tri-state" device. As you can see on the "Function Table", there are three states for the output pins on the L293. "High" indicates that the output pin is attached to the supply voltage, making that pin a "source", supplying positive voltage to a motor or circuit. "Low" indicates that the output is attached to ground, allowing the pin to "sink" power, connecting a circuit or motor to ground. "High-Impedance" effectively means that the pin is not attached to anything. Technically, of course, it is attached, but the impedance (similar to resistance) is so high that only an insignificant amount of current will flow through the pin). This comes in useful for a number of different applications, such as controlling whether the motor will act as a "brake" when turned off, or allow the driveshaft to turn more freely. More information is contained in the datasheets for both the L293 and the PIC.

NE PACKAGE
(TOP VIEW)



FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low-level
X = irrelevant
Z = high-impedance (off)

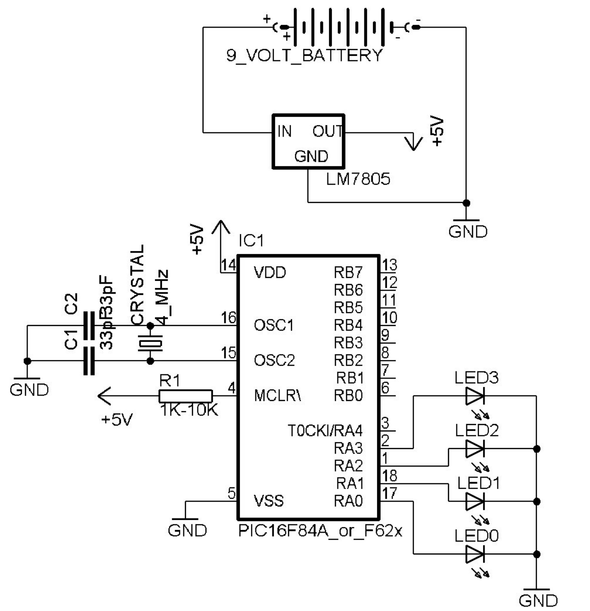
L293D Connections and Logic

Since our circuit will not make use of the tri-state abilities of the L293D, we can connect pins 1 and 9 to the +6v supply from the 4xAA battery pack. We can also connect both the power voltage supply (pin 8) and logic voltage supply* (pin 16) to the +6v supply. This gives the chip two-fold symmetry, making it very easy to hook up. Essentially:

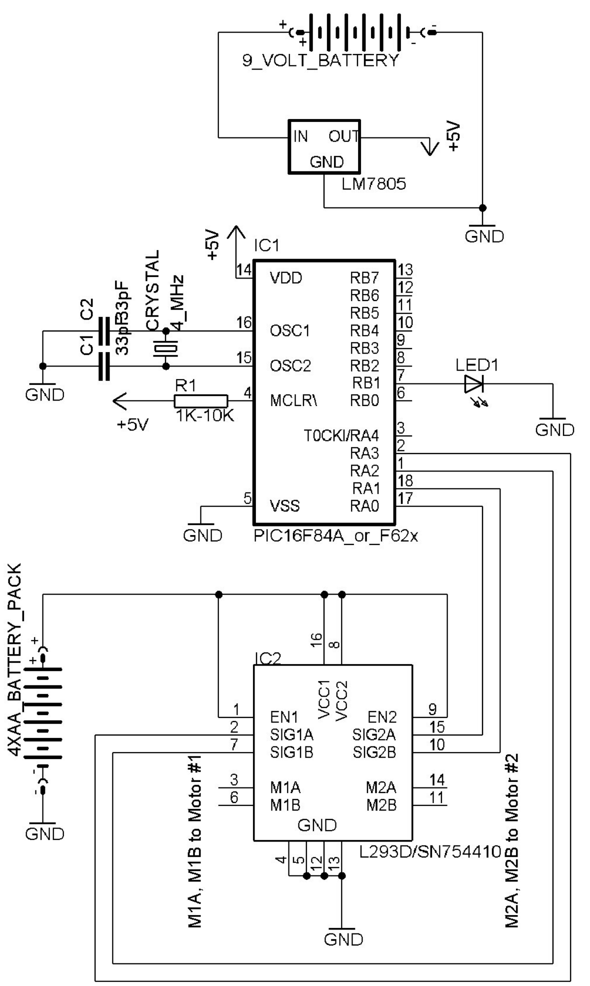
Each corner pin is connected to the +6V supply from the 4xAA battery pack.
Each pin one in from the corner is connected to an output pin from the PIC
Each pin two in from the corner is connected to a terminal on a motor.
The four middle pins are connected inside the chip. One of them needs to go to ground.

Note that while there are two voltage sources (+6 volts and +5 volts), they must share a common ground in order for this circuit to work. (ie. the black wires from each battery need to be connected to the same line on the breadboard in one way or another.)

* The L293D can handle voltages up to 36V for the motors. The logic inside the chip, however, can run at a much lower voltage. Typically the power supply for the motors will be at a much higher voltage than the logic supply for the circuit board. In our case, however, when the motors are running the logic voltage (powering the PIC) will be higher than the power voltage. Connecting the +6V supply to all four corners of the L293 isolates the PIC from this lower voltage.



Schematic For Test Circuit #1 – LED Output Only



Schematic for Test Circuit #2 – Motors Attached

A Test Program for Both Circuits.

```
processor 16f84
__config _XT_OSC & _WDT_OFF & _PWRTE_ON
include <p16f84.inc>

COUNTER1      equ    h'20'
COUNTER2      equ    h'21'
COUNTER3      equ    h'22'
DELAY         equ    h'38'
       movlw  0
       tris   PORTA
       tris   PORTB
START:
       movlw  b'01010101'
       movwf  PORTA
       movwf  PORTB
       movlw  DELAY
       call   PAUSE
       movlw  b'10101010'
       movwf  PORTA
       movwf  PORTB
       movlw  DELAY
       call   PAUSE
       goto   START

PAUSE:        movwf  COUNTER1
PAUSE1:       movwf  COUNTER2
PAUSE2:       movwf  COUNTER3
PAUSE3:       decfsz COUNTER3
       goto   PAUSE3
       decfsz COUNTER2
       goto   PAUSE2
       decfsz COUNTER1
       goto   PAUSE1
       return

end
```

This program should cause the LEDs to blink in circuit #1, and the motors to alternate direction (and LED1 to blink) in circuit #2. By changing the binary values placed in PORTA, and the DELAY value stored in W before calling PAUSE, you should be able to make your robot move on it's own.

Notes:

- You may need to modify the first four lines to suit your particular processor.
- The __config begins with two _ characters.
- Get the LED's blinking with circuit #1 first, then try circuit #2.

Schematics created in EAGLE Layout Editor. A freeware program with PCB layout and autorouting. www.cadsoft.de