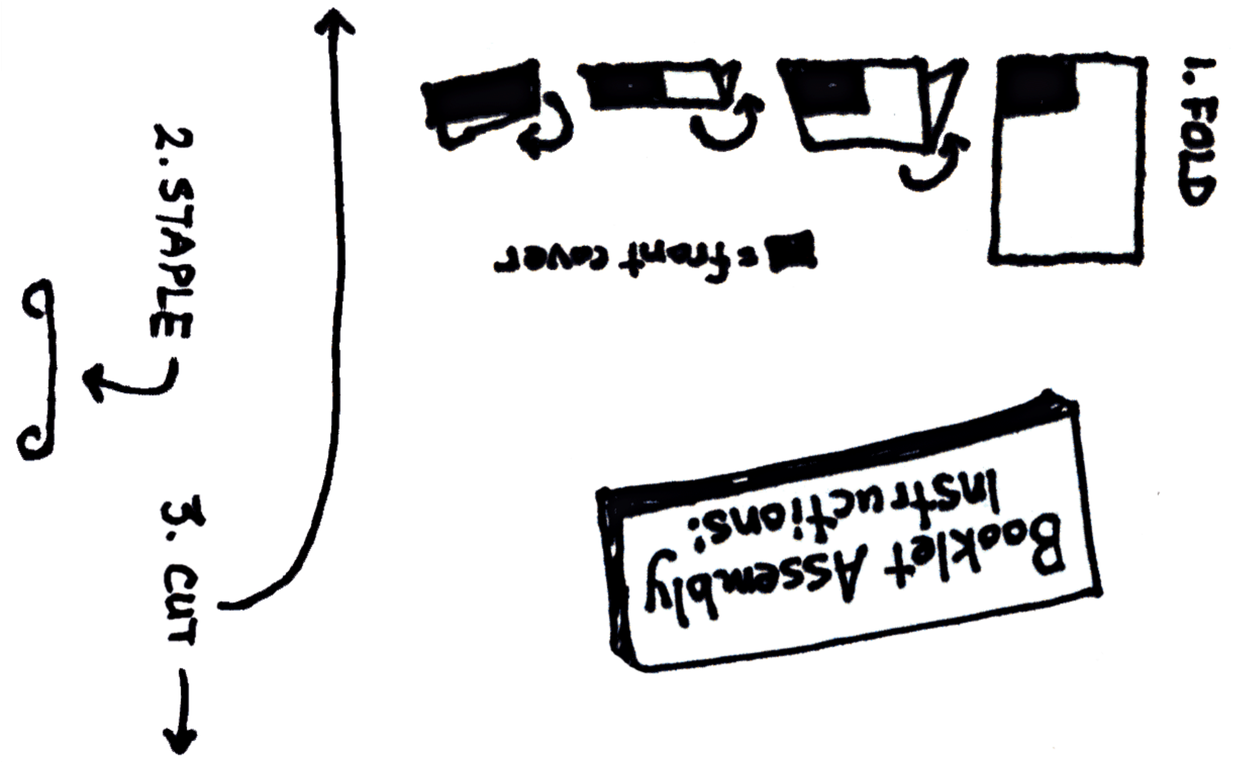
(fly)light

D.I.Y. Guide #1

INTRODUCTION TO PIC PROGRAMMING
(USING THE PIC16F627)



Pin I/O

The 628 has two ports: Port A and Port B. There are 8 pins on each port plus one pin (14, marked Vdd) for the power source (positive on an A/C adapter) and another pin (5, marked Vss) for ground (negative).

If you look at the pin diagram, Port A is at the top marked with RA0 (pin 17), RA1 (pin 18), RA2 (pin 1), etc. Port B is at the bottom marked with RB. To target one pin with code, you would use it's pin name: PORTA.0, PORTB.0, etc. As you can see, the physical pin number doesn't need to correspond with the pin name.

The pin diagram also denotes the multiple functions of a pin so refer to the pinout description table for a brief overview of what the pin can do and how it should be used. For example, pin 17 can act as an I/O pin (RA0) or as an analog comparator input (AN0).

Since (fly)light is really only turning on or off LEDs, I'm looking for the output type CMOS in the table. For Port A, the CMOS output pins are 17, 18, 1, 2, 15, 16 which correspond to PORTA.0, PORTA.1, PORTA.2, PORTA.3, PORTA.6, PORTA.7. For Port B, all pins (6, 7, 8, 9, 10, 11, 12, 13) can be used as CMOS outputs (PORTB.0, PORTB.1, ..., PORTB.7).

What about pin 3 (PORTA.4)? You can still use this as an output pin but it's an open drain (OD) type of output. With a CMOS output, it's either fully on or fully off. With an OD output, the power level sort of "floats" so you need to use a pull-down resistor. The basic idea of a pull-down resistor is that, given no other input, the pin will default to an "off" value instead of doing wacky stuff. After using a pull-down resistor, you can use pin 3 just like a CMOS pin.

And pin 4 (PORTA.5)? This pin is set aside for reseting the chip or as an input pin -- so no output capabilities at all.

* $2.84 CDN for the PIC16F628-04/P when ordered through digikey.ca in multiples of 25. I don't know if these are the best prices around but it's nice that Digi-Key incorporates any duties, tariffs and brokerage fees related to cross-border shipping into the prices listed in their catalogue... I personally hate paying extra when getting packages in the mail.

It also has built-in serial communications abilities so it's possible to have a computer control the hardware attached to your PIC. For (fly)light, this might translate to computer-driven, LED animations.

The first and foremost reference for figuring out a PIC's capabilities is it's datasheet. The PIC16F62X datasheet can be found on the manufacturer's website (microchip.com) or from the retailer in which you bought your chips from.

The .pdf datasheet is 170 pages long but don't worry... you don't need to read all of it (or even understand 95% of it) to get started. Concentrate on the pin diagrams on page 4 and the table of pinout descriptions on pages 11-12. (See next pages for an excerpt.)

This hex code needs to be transferred to your chip using a PIC programmer. microEngineering Labs also offers the EPIC Plus Programmer which is a specially designed circuit with a spot to insert your PIC and it comes with operating software.

After installing and running the EPIC software, you'll have to make a few program settings:

Almost there...

The PIC doesn't understand BASIC so you need to compile the BASIC code into machine language. You can either buy the PicBasic Pro Compiler from microEngineering Labs (melabs.com) and install the software on your computer. Or try an online compiler that you purchase a subscription for (compilespot.com offers a full demo version).

Either way, after you compile the BASIC code, you'll get a .hex file that contains mysterious computer jargon.

PIC16F62X

For the purposes of (fly)light and learning, we're going to specifically use the PIC16F628 model. Note that there is also a 16F627 chip that is almost exactly the same as the 628 except that the 628 has room for more lines of code and costs a few more cents than the 627.

The 628 might be overkill for lighting up LED matrices one row at a time (which is what (fly)light is doing right now) but it's best to start simple. You'll be able to build on what you learn here for later projects.

The 628 is a great beginner's PIC for several reasons including price -- typically $5 for one or $3* if you buy in bulk.

It has an internal 4mhz oscillator which means you don't need to buy an extra clock crystal in order to get it up and running. Or if you want it running faster, you can add a 20mhz oscillator.

1) Select PIC16F628 from the drop down menu.
2) Under Configuration > Oscillator, choose INTRC (IN). This will select the 628's built-in clock.
3) Uncheck Configuration > Master Clear Reset Enable if you're not going to use a reset button in your circuit. This will save you from having to solder in another pull-down resistor.
4) Uncheck Options > Update Configuration otherwise you'll have to select INTRC (IN) every time you open a new .hex file.

Then to program your PIC:
1) Connect your PIC programmer to your computer using a parallel port computer cord and plug into a power supply or use 2 – 9V batteries.
2) Put your PIC chip into the PIC programmer circuit board.
3) Open your .hex file using File > Open.
4) Then select Run > Program.
5) You should get an okay message if everything went well.
6) Take out the PIC chip and put into your circuit!