IPC Graphics and Electronics PIC stuff

IPC has designed a few ‘Turn Key’ Pic products, from trash compactors to FM transmitters. We typically
a customers wish list, generated a design document with prices and dates, selected components, drew the
schematic, designed a PWB, created BOM’s, drawings and test procedure for the Contract assemblers. We
monitored the finished design through the first few production runs and taught the assemblers how to test a
debug the product.

These projects include: Trash compactor controller, Diesel engine monitor, FM transmitter with solid state
continuous loop audio, Async converter for Hitachi LCD drivers, Linear Actuator controller, process test
equipment for the paper industry, MIDI Light dimmer, and a Pinewood derby timer.

We designed the following two example designs a few years ago. They are finished designs that have prov
themselves in the field, but they are not exactly beginner friendly. They are intended to provide a seed or a
starting point from which a designer can create their own project. If you are comfortable with an Oscillosc
or a logic analyzer, and better with C than English, then these are ready to use. Unfortunately we have no
PCBs, parts or kits available.

PIC 16C63 Midi controlled Light dimmer.

This is Ver2.0 of a finished project.

I usually do projects about the same way every time. Product definition (10 min), Code outline (10 min),
Schematic generation, Build prototype, Write code all at one sitting (This one took about 6 hours), Take a
of the above file, and delete all but a small section. Debug the section, and put it back into the original file.
Repeat with other chunks of code until done. Test the program as a whole, Mass produce and be famous (n
got to this last step yet).

Files in MIDI.ZIP

MIDI20.C Source code for Midi Light dimmer

MIDI.SCH Schematic in OrCad Version IV (Dos version)

MIDI.LIB Library file for OrCad Schematic

MIDI.DXF Schematic in DXF Format

MIDI_SCH.PDF Schematic in Postscript format.... Copy to postscript printer

Download "MIDI.ZIP" PIC16C63 C source code and schematics.

View Schematic for the MIDI light dimmer (PDF File).

View Photo of finished product.

Use the terms Port, Channel, Device, and Box like this. You can have lots of MIDI ports (Connectors) on a
machine. Each MIDI PORT can have 16 CHANNELS of 127 DEVICES (notes) each. Each light dimmer b
consumes 5 DEVICEs: 4 lights and one relay.

As it is, the box has a MIDI interface. There's an input connector and an output connector. The boxes' outp
connector goes downstream to the next boxes' input connector. There's a DIP switch in each box to select
which of the 16 channels the box will respond to, and what device the first light will respond to. The four l
and the relay will respond to four consecutive device addresses. So... With the dipswitch, you can set a box
be on channel 5, and use devices 32 thru 36. The first light will respond to device 32, the next light device
the next 34, and the last 35. The relay will then use device 36. You can connect another box to the first box
output connector, and set it to respond to channel 5 devices 37-41, (or any channel, any device).

Cakewalk is a MIDI sequencer / editor and it is good. You can just draw the desired intensity of a particula
light by drawing a volume envelope of a single note. If you're in a stage act, and already have Cakewalk do
the sound, that's the way to go. The existing design can be modified to use standard baud rates, and the RS
Electrical interface. Then you can just fire off commands via RS-232. Cakewalk would be a little difficult t
interface to if you just wanted to make a home lighting system that responds to opening doors and time of
For that you just wip something up in Visual basic using standard com port commands.

PIC "Pinewood Derby" style race timer.

PIC based "Pinewood derby style" race track timer. One switch closure starts the race and initiates end of r
process. That switch is mounted to the starting gate. When the gate snaps open, the cars start rolling. The g
stays open until the race is over. The act of closing the gate, clears the timers, sets the state machine, and
process flags to be ready for the next race. Digital inputs to indicate finish line crossings, one for each of fo
lanes. A digital high is output by the track when a car shadows the light detector at the finish line. This tim
detects the rising edge of this signal. 11 bits of output are consumed by the 8 bit interface to a 2 line, 40 ch
LCD. There's plans for a RS232 output to squirt the race times out a serial port. The timer is a great use du
car tune-up. It doesn't seem to add much to a race.

A MICROCHIP 16C63 Micro controller is used. The main clock oscillator is a 4.0 MHz crystal. It divides
nicely to even nanoseconds for timing, and uses the slightly cheaper versions of Pics.

The program is written in C using the CCS compiler. In general, the program runs from a main loop. The l
polls the start switch, builds the text for the LCD, refreshes the LCD, and will send characters out the seria
port. Interrupts are used to handle finish line crossing information from the track. The interrupt routine tak
snapshot of a counter value for each lane upon each RB interrupt.

Download Pinewood timers' PIC16C63 C source code.

Download Pinewood timers' OrCad schematic.

View Schematic for the Pinewood timer.

"Pinewood Derby" style race Finish line car detector.

Detecting the cars crossing the finish line turned out to be a little more than I planned. The first system use

that when one flipped, it would prevent the others from flipping. I mounted the infrared detectors and lamp
across the track from each other. When the car wheel broke the beam, the detector sent the signals, that flip
the flippers that finally lit the winner led.

I was worried about the life expectancy of the bulb, and wasn't satisfied with detecting wheel crossings to
decide winner, so version two was made. The cars are lined up at the starting gate by a dowel against the fr
of each car. Since the cars are lined up by their noses, the finish line detectors should look for the cars nose
not their wheels.

The old way gave stubby cars the edge, this new way gives blunt cars a slight edge. Overall it seems more
There is some thinking going on about removing the winner led from the track, and putting them into a han
held box that only the judge can see. We currently have a human judge to provide the deciding vote on wh
wins. It may save a little embarrassment, and may prevent a few arguments in the case where the electronic
detector says one thing, and the judge sees another. The judge could weigh the electronic detectors answer
along with their other input when deciding on a winner.

I tried several light sources before I found one that worked well. I first tried little focused lamps from Radi
shack. They were pure junk. Out of 16 lamps, I found only two that pointed straight enough to use. I tried
LED's. They didn't have enough range. Lamps out of MAG flashlights were bright enough, but burned out
about 6 hours. I finally tried Infrared LED's. They work great. Plenty bright enough to shoot the 6 inches f
the light bar and down to the track, and they've been on for a week now, and will probably last forever. I h
mount little red leds near the detectors to indicate when they aren't getting enough light. They're needed
because you can't see where the leds are shining. But with nearly a three inch circle of illumination, it's rea
easy to align them.

Download Pinewood finish line detector OrCad schematic.

View Schematic for the Pinewood finish line detector.