LocoBuffer Documentation

Subjects

Disclaimer
Overview
Hardware
Software
Revision History
PIC Programming
Debugging Tips
Printed Circuit Board
ICSP (In Circuit Serial Programming)
Assembly instructions
Contacting me

Disclaimer
This is here to protect myself. This device has no purpose what
so ever. Use it at your own risk. It is not Loconet certified.
Loconet is the property of Digitrax. I'm not an electrical engineer.
Blah, Blah ....

Overview

I renamed LOCOMODEM to LocoBuffer because it's more of a buffer
than a modem. LocoBuffer is a hardware device that provides
a hardware interface between a Loconet and a RS232 serial port.
It provides all the timing necessary to interface the 2. This allows
multi tasking operating systems the ability to deal with the device.
As long as the computer can read and write to a RS232 serial port
and can service requests about ever 1/4 second it will be able to
use the device.

Hardware
There is a parts list posted in the files section of this list.
Current retail value of the parts necessary to build the device
is about $35.00.
Summary of what it is:
PIC16F873, MAX233CPP, LM339M, 7805 and some connectors and some
small parts.
All the parts are available from Digikey and I will program the 16F873 for
free if you don't have a programmer.
I used 1/4 watt resistors but 1/2 watt will work.
MAX233CPPA will also work (but is $3 more).
I have designed this device using a oscilloscope and the wave form this
device puts out matches the wave form put out from a DS54.

What it does
Takes Loconet in and buffers it and sends it out the serial port at 16457, 19200 or
57600 baud (jumper selectable).
Takes serial in at 16457, 19200 or 57600 baud and buffers it and puts it out on the
Loconet. It handles all timing, including CD Backoff and break in case of a
collision. It also does them both at the same time.
The latest PIC hex code echo's the Loconet packet that the device sends out. This was requested
so that a program knows what order the packets were put out on the Loconet.

Software
LocoBuffer should works with any Wintel OS.
LocoBuffer should work with any other OS including MAC and Linux. I don't
have a MAC so I couldn't code anything and test it and I don't have the time
to hook it up to my Linux machine and code up a program and test it.
I also have a simple Visual Basic program that demonstrates it's
functionality. This is posted in the files section of this list.
If you have VB on your machine, all you need is RRCntl_source.zip. Copy it
to a directory and unzip it, and then bring it into VB and do a make .exe
If you don't have VB on your machine then you need RRCntl.zip. Copy
it to a directory, unzip it and run the setup program. It will install
a few Microsoft Visual Basic DLL's and the RRCntl.exe executable.

If you write your software it must be able to do the following.
Open up the serial port at 19200 8,n,1 or 57600 8,n,1
Read or write binary data to the serial port.
That's it, no timing to worry about. The data you get will be full packets.
The binary data will be packets that are 2,4,6 or multi byte in length.
They will contain the data as documented in the Digitrax Loconet Personal
Edition 1.0. Currently the device buffers 32 bytes input and 32 bytes
output. All my current testing indicate that this is enough but this will
be expanded in the future to allow for slower PC's.

Revision History
Version 1 - original release
Version 1.1 -
Version 1.2 - Added echo
Version 1.3 - Added switch selectable echo
Version 1.3.1 - Added Pin 4 LED to light when Loconet traffic coming in
Version 1.3.2 - Added Pin 4 LED to light with all Loconet traffic
Version 1.4 - Added MS100 compatability mode
Version 1.4.1 - Added support for ICSP
Version 1.4.2 - Fixed ICSP programming configuration byte
Version 1.4.3 - Fixed 17 byte packet problem
Version 1.4.4 - Fixed a cdbackoff timing problem
Version 1.4.5 - Fixed a possible checksum 00 problem
Version 1.4.6 - Fixed a problem caused by V1.4.5 (effects DecoderPro)
Version 1.4.7.1 - Fixed a problem dropping a byte in a long packet

PIC programming
If you want me to program a PIC for you, send me an email and I will
send you my address. There is no charge for this, just send me a 16F873
PIC and return postage and I will program it and test the PIC for you.
or
If you have your own programmer, you want to use the following settings:
Oscillator - HS
Watch dog timer - Off
Power up - On
Code protect - Off
Brown out - On
Low voltage programming - Enabled
Data EE protect - Off
Flash program write - Disabled
or
Use ICSP (In Circuit Serial Programming)

Debugging tips
If you have trouble with getting the circuit to work
try the following:
Check the wiring
Check the wiring
Check the wiring (just kidding)

Using a simple 5 volt logic probe (one that can stretch a pulse),
you should see the following signals on the PIC.

Pin 2 (Loconet out from LocoBuffer)
should be low until the pc sends a loconet packet
out the serial port and then it should flash.

Pin 3 (Loconet in from Loconet)
should basically be high and flash whenever loconet
packets are present.

Pin 17
With echo on
should be high and should flash whenever the LocoBuffer sends
data out to the serial port that originated from the pc
should be high and should flash whenever the LocoBuffer sends
data out to the serial port that originated from the Loconet

With echo off
should be high and should flash whenever the LocoBuffer sends
data out to the serial port that originated from the Loconet

Pin 18
should be high and should flash whenever the LocoBuffer receives
data from the serial port that originated from the pc

Pin 22 should be low almost always.

Pin 23 should be low until the pc sends a loconet packet
out the serial port and then it should flash.

Transistor problems
The transistor I used is a 2n2222 because it works and is
available at Radio Shack. Other people have used a bc337.
I couldn't find them here in the US but it should work.
Any transistor (npn I believe) with enough gain should work.

Your serial cable should be straight through. I crossed
pins 2 and 3 in the device to eliminate the need for a
null modem cable.

Lets start with the Loconet. I assume you can generate
Loconet packets. If you have a DT200 it only generates
Loconet packets with turning power on and off to the track.

First test - check Loconet to serial port path
Using the logic probe, do you see flashing
on pin 2 of the LM33N when a packet is on the Loconet?
You should.

Then do you see flashing on pin 3 of the PIC?
You should.
Then you should see flashing on pin 17 of the PIC.
Then you should see flashing on pin 18 of the MAX233.

Second test - check PC to Serial port
Start up the RRCntl program
Set the com port properties to either 19,200 or 57,600
and to the correct com port. (by the way - the original
schematic had the jumper backwards for the baud rate)
Click on the open port button
Open the engine menu
Click on get engine
You should see flashing on pin 19 of the MAX233.
Then you should see flashing on pin 20 of the MAX233.

Third test - check serial port to Loconet
Next click on the speed bar.
You should see flashing on pin 18 of the PIC.
You should now see flashing on pin 2 of the PIC
You should then see flashing on pin 3,4 of the
Loconet out RJ12 connector.

Printed Circuit Board

Printed circiut boards are available from me. They are currently
$7.10 (US dollars) plus shipping. Kits are also available from me.

The printed circuit board and schematic were designed using
Eagle circuit design software. If you know how to make circuit
boards, don't read the rest of this section. If you don't then
read on.
After you make the board you need to drill it out. I recommend
using different size drill bits for the IC holes and the resistor
holes. I also recommend you use 1/4 watt resistors. They are the
correct length.

ICSP (In Circuit Serial Programming)
ICSP will program a new PIC or one that already has code in it
(it will overwrite the existing code).
To use ICSP you must build the LocoBuffer to the latest schematic
and code Version 1.4.1. or higher.
There are 2 extra wires (PIC pins 28 to 22 and pins 27 to 18)
and 2 extra jumpers (JP4 and JP5).
If you have lower level code in your PIC, that's OK, just upgrade
it using ICSP. You will also be able to program LocoIO when it's
out.
I have included a Visual Basic program that will program any hex
code into a PIC 16F873 chip.
To program the PIC do the following
Build LocoBuffer with the latest schematic.
Download isp.zip and unzip it. This program has been installed and
Tested on Windows 98 and Windows 2000 Professional. It should
work on other Windows 9x and higher systems but hasn't been tested.
Run setup from the unzipped files This will install some Microsoft
Visual Basic .DLL's and .ocx's and the ISP program and a hex file.

Download the latest hex file from the list.
Connect your LocoBuffer to your pc's serial port.
Connect power to the LocoBuffer.
Run isp by double clicking on it.
Follow the instructions on the screen which are:
        Set JP5 on the LocoBuffer to program mode.
        Using CommPort Properties Menu point to correct comm port
        Using CommPort Menu select Port Open
        Set JP4 jumper to program mode
        Using File Menu select Open Hex File
        Using CommPort Menu de-select Port Open
Test the LocoBuffer.
The trick to using the programmer is that you first open the
comm port to bring DTR and RTS to the correct level and then
bringing JP4 from normal mode to program mode. Also when you
connect the jumper to program mode, try to put it on only
once (debounce the switch) if you touch it a couple of times,
put the jumper back to normal and then to program (this sounds
harder than it is).
Note that the comm port speed isn't important for ICSP to work.
Note about the isp program.
        I have to add some code to allow you to change the short wait
and long wait parameters. Timing isn't critical but there are a
few requirements. The values I have set now work on a 500 MHz
Pentiun III and slower machine. They may work on faster machines.
On the 500mhz machine, programming takes about 20 seconds. If your
machine is half the speed then it should take about 40 seconds.

Assembly Instructions

First identify all the parts.

CAUTION – The PIC, Maxim 232 driver and the LM339 op amp are static sensitive devices. Don't remove them from their packages until you are ready to install them and then don't touch the pins on them. Also make sure you discharge yourself before you handle them. It is best not to work on carpet when you are handling these devices.

Assembly order is not important but you might want to follow the order below. Check off each item as it is completed. Insert the part, solder it in then inspect the solder joint to make sure it is shinny and complete. Then cut off the excess lead.

Soldering this project is rather easy as long as you have the correct tools and are neat. You should use a small soldering iron, 15 to 25 watts. Also use the thinnest solder that Radio Shack has. Make sure you are using solder for electrical parts and **don't** use acid core solder under any circumstances.

All statements relating to orientation are with the top of the board up and the word LocoBuffer in the left lower corner.

First put in the resistors. They can go in any orientation. They are all 1/4 watt 5% resistors. The resistors are the cylindrical shaped parts with leads on each ends and colored bands around them. The bands are indications of the resistance of the part. If you have good eyesight you can use the bands to identify the resistors. If not use an ohm meter to measure the resistance of them.

R8 – 1K resistor – brown black red gold

R9 – 1K resistor – brown black red gold

R5 – 47K resistor – yellow purple orange gold

R6 – 47K resistor – yellow purple orange gold

R4 – 100K resistor – brown black yellow gold

R3 – 220K resistor – red red yellow gold

R2 – 22K resistor – red red orange gold

R1 – 10K resistor – brown black orange gold

R7 – 4K7 resistor – yellow purple red gold

All though the IC sockets can go in either way, convention says to put the notched end of the socket pointing to the #1 pin. The board is marked for the #1 pin.

14 pin socket

20 pin socket

28 pin socket

Don't put any IC's in the sockets yet

B1 - DB101MS-ND Bridge rectifier – Black square IC with 4 pins - The orientation of this is important. The AC pins go to the left and the minus (–) is in the upper right corner and the plus (+) is in the right lower corner. If this is backwards you won't get +5volts out of the device.

F25 - DB25 pin female connector

JP1, JP2, JP3, JP4, JP5 - 5 headers – cut the headers into groups of 3. Make sure you don't accidentally put one in the holes for IC4 (the 7805).

J1 – RJ12 6 pin Loconet connector

Q1 – 20 mhz ceramic resonator – Small square brown component with 3 pins coming out the bottom. This can go in any orientation.

IC4 – 7805 voltage regulator – large black rectangular device with 3 heavy leads coming out the bottom. Orientation is important. The words must point to the right. If this is backwards you won't get +5volts out of the device.

C1 – 100uf electrolytic capacitor – Black round cylinder with 2 leads coming out of the bottom. Orientation is important. One lead has a minus sign (-) on it. The minus must be placed facing the word LocoBuffer. If this is backwards you won't get +5volts out of the device.

D1 – Red LED – Insert with the long lead to the right.   When power is applied, this should light.

D2 – Red LED – Insert with the long lead to the left. This shows Loconet Traffic.

T1 – NPN transistor – small black round device with 3 leads coming out the bottom. Looking from the top, this is a D shaped device. Orientation is important. This is the part that will most likely be installed incorrectly. The flat side of the D should be facing the RJ12 Loconet connector. The center lead goes in the hole with the B (for base) on it.

Before you install any of the remaining IC's you should put power into the device. You connect a wallwart to the connections labeled Pad1 and Pad2. You can put in either AC or DC. It should be a power supply with between 7 – 15 volts and at least 100ma. Plug in the wallwart and connect the black lead of a voltmeter to pin 8 of IC1 (the PIC) and connect the red lead of the voltmeter to pin 20 of IC1. The volt meter should read about +5volts if it's less than this then you may need a more powerful wallwart. IC1 is numbered from 1 to 14 on one side and 15 to 28 on the other side. You can now check that all the IC sockets have the +5 volt on the proper pins.

Important – Remove power before completing the next steps.

Next install the jumpers. Jumpers JP1, JP2 and JP3 should probably be set to 2 and 3
And jumpers JP4 and JP5 should be set to 1 and 2.

Now install the remaining IC's. CAUTION – these are static sensitive devices. Read the warning at the top of the page.

IC1 – PIC processor. Orientation is very important. Pin 1 on the PIC must line up with pin 1 on the socket. Pin 1 on IC devices is either marked with a small indent over pin 1 or pin 1 is on the left side of the end if the IC with an indent or notch in it.

IC2 – MAX233 RS232 driver. Orientation is very important. See note above.

IC3 – LM339N op amp. Orientation is very important. See note above.

Congratulations. You have built a LocoBuffer. Now lets see if it works.

Connect a straight through RS232 cable between the LocoBuffer and your computer. Connect a straight through active Loconet cable to the LocoBuffer. Plug in the wallwart.
Startup the sample RRCntl program on your pc. Set the serial port to 57,600 baud and to the correct comm port. Then open the port. Now run your trains with either the RRCntl program or your regular throttle. You should see all the Loconet packets in the window. If you don't then go to the troubleshooting section of the LocoBuffer documentation.
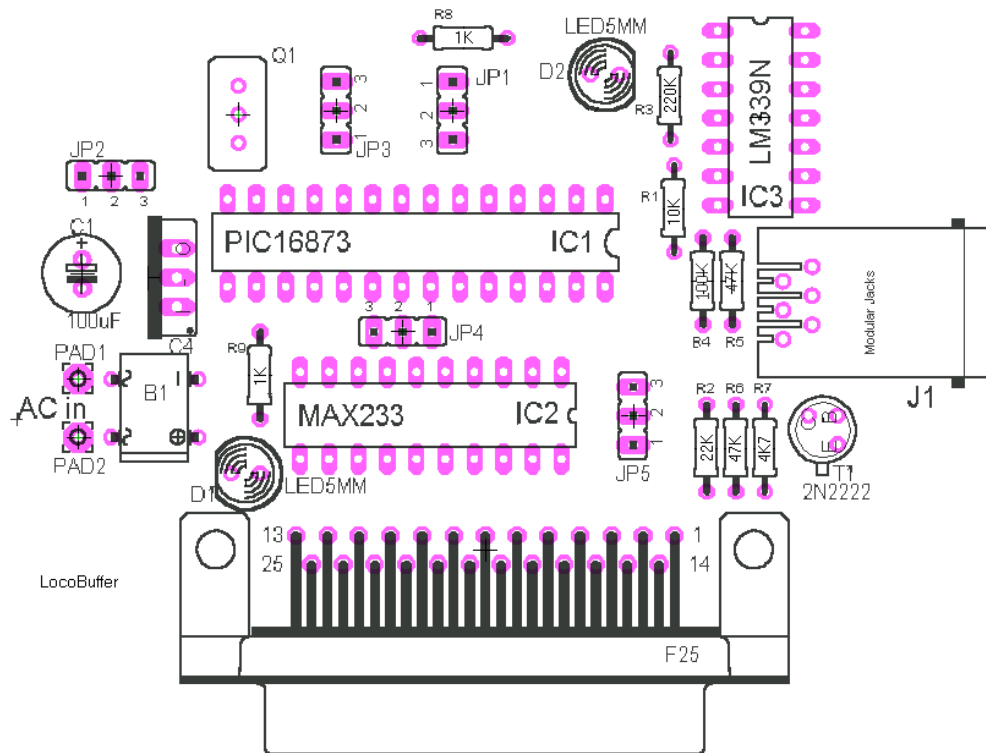
JP1   1 – 2 19200 Baud
           2 – 3 57600 Baud
JP2   1 – 2 Echo off
           2 –3 Echo on (recommended)
JP3   1 –2 MS100 Compatibility Mode
           2 – 3 LocoBuffer Mode
JP4   1 – 2 Normal Mode
           2 – 3 Program Mode
JP5   1 – 2 Normal Mode
           2 – 3 Program Mode

Contacting me

I'm available to give you help by E-mail.
John Jabour
jjabour@optonline.net