# LocoIO

**Subjects**

Disclaimer
Overview
SV's
Revision History
PIC Programming
PC boards and kits
Functionality
Assembly Instructions

**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. I'm a hobbyist and developed LocoIO for my own use. I find it useful and am sharing it with the model railroad hobbyist's. I have used it on my systems without causing any damage. This is not to say it will work in all environments. Use your own judgement.

**Overview**

LocoIO is a hardware device that is a building block for people wanting to utilize the Loconet. LocoIO is the motherboard, it requires additional circuitry to do productive work. It uses the same basic design as LocoBuffer. It is configurable by setting values into SV registers in the PIC processor. There are 16 real ports that can be configured and in the future there will also be virtual ports and more functionality (including turntable control). Basically what this device does is converts a 0 or +5 volt signal to a Loconet message and/or converts a Loconet message to a 0 or +5 volt signal. It also does software de-bounce of switch input.

Currently this would allow you to build a remote CTC panel that would contain switches and/or LED's or would allow you to control a signaling device. When a block becomes occupied or empty a LED would come on or off. Which block and which LED is configurable. When you throw a switch it will operate a remote turnout. This is also configurable. You could also have a LED turn on or off depending on the direction that the switch is thrown. Since all this is done with Loconet messages, the LED's and the switches don't have to be on the same LocoIO CTC panel. An example appears further below.

All values in this document are decimal unless preceded by a 0x. When you look at packets using the debug screen, all values will be hex. There are 2 documents that address the LocoIO device. This is one of them and the other is the SV Tutorial.pdf which is also available on the LocoBuffer web site www.locobuffer.com .

**Programming SV's**

The SV's in LocoIO can be configured using the RRCntl program, which can be found at on the LocoBuffer website www.locobuffer.com . You must use Version 3.1 or higher. You can also write your own program to do this. Programming specifications for this are given in this section below. Currently the SV's are configured using Loconet OPC_PEER_XFER messages. In the future, I hope to add programming using a DCC packets on a programming track.
Note: Make sure LocoBuffer is at Version 1.4.7.1 or higher to minimize the problem mentioned below.

**Commands for setting SV's using RRCntl**

01 – SV write
02 – SV read

**Commands for setting multi ports using RRCntl**

03 – multi port write
04 – multi port read

**Current addressing**

The PC is 01 80
The default LocoIO is 01 81 (you will have to change this if you have more than 1 LocoIO). Digitrax has assigned the 01 high address for LocoIO devices.
Note: A broadcast packet can be sent out using a value of 0 in the DSTL Destination low address field. This allows you to set the SV1 value to an initial value (assuming you didn't get the PIC programmed from me) or to fix SV1 if you accidentally changed it to an unknown value.

**Loconet Program Packet Layout (from PC to LocoIO)**

In order to program the SV's you use Loconet Peer to Peer messages (OPC_PEER_XFER). The syntax of this message is documented in the Loconet Personal Edition 1.0. This document will address the field's LocoIO uses.
0xE5    OP Code
0x10    message length
SRCL    Source low address
DSTL    Destination low address
DSTH    Source and Destination high address
PXCT1
D1      Command
D2      Register
D3
D4      Data
PXCT2
D5
D6
D7
D8
CHK     Checksum

Example
E51050510100002010000000000000009 (all values in this example are hex)
This is sending out a read command to LocoIO # 81 (0x51) and requesting the data from SV number 1.

**Loconet Program Packet Layout (reply from LocoIO to the PC)**

0xE5    OP Code
0x10    message length

SRCL    Source low address
DSTL    Destination low address
DSTH    Source and Destination high address
PXCT1   High order bit of Version
D1      Original command
D2      SV number requested
D3      Lower 7 bits of LocoIO Version
D4
PXCT2   High order bit of requested data
D5
D6      Requested Data
D7      Requested Data + 1
D8       Requested Data + 2
CHK     Checksum

Example
E510515001040201070008005101014012 (all values in this example are hex)
This reply is from LocoIO # 81 (0x51) and saying that SV 1 has the value 81 (0x51)

**LocoIO Setup Byte**

SV0 is a setup byte that does global configuration of the LocoIO.  Current values are:
Bit 0 – 0 = Normal RAM refresh, 1 = No RAM refresh
Bit 1 – 0 = Normal LocoIO,          1 = LocoIO is for turntable control
Bit 2 – 0 = Detectors active high,  1 = Detectors active low

**Note: If you change SV0 you must cycle power to the LocoIO for it to take effect.**

**Configuration byte**

All ports use 3 bytes to configure them.  This will be called a SV set.  All SV sets start with the first byte
being a configuration byte.  Below are the values that can be in the configuration byte.

Bit 7 –   0 input
          1 output
Input     Command to send out
          Currently 15 (0x0F)
                   47 (0x2F)
                   111 (0x6F) are available
Output    Command to look for
          Currently 192 (0xC0)
                   128 (0x80) are available

**Programming SV's**

**Note: In the examples below you will see CC at the end of each packet.  This means the checksum
value.**

**Note: I recommend that you print out a copy of the SV page of this document and keep a record of
what values you have in which SV's.  Although using ISP to re-program your PIC will not overlay
your SV's, you should still keep a paper copy.  You can also use RRCntl to write the SV values to a
file so that they can be reloaded at a later date.**

There are currently 4 sets of values that SV's can deal with.  They are

          Generate a OPC_SW_REQ (0xB0) message (usually from a switch action)
          Generate a OPC_INPUT_REP (0xB2) message (usually from a block detector)
          Receive a OPC_INPUT_REP (0xB2) message (usually turns on a LED)

Receive a OPC_SW_REQ (0xB0) message (usually turns on a LED)

**To generate an OPC_SW_REQ (0xB0) message (usually from a switch action)**

To program a SV to be an input SV you will do the following.

The input configuration value is any of the following values
15 (0x0F) – toggle switch
47 (0x2F) – push button switch (activate low)
111 (0x6F) – push button switch (activate high)
**Note:** LocoIO does software de-bounce of switch input.

**Caution: Do not leave inputs floating. They should either be at 0 volts or +5 volts. Leaving them float can cause unpredictable results.**

If a switch is connected to the input, determine what data you want to send with the command. The command that will be sent is the OPC_SW_REQ (0xB0). The data will be the address of the device you want to activate and then the value 2 which is 16 (0x10). The address can be determined by activating that device with a DT100 type throttle and then subtracting 1 (one).

**Example 1**
Toggle switch
The DT100 type throttle is put in switch mode and then turnout 29 is selected. Then the closed button is pressed and turnout 29 is closed. To have this turnout work off of a switch connected to LocoIO you would put the following values in the proper SV's. Lets assume that the switch is connected to pin 17 on the JP1 connector of LocoIO. Looking at the SV chart we see that SV's 27 – 29 apply to JP1 pin 17.
SV27 – 15 (0x0F) – This is the configuration SV for JP17
SV28 – 28 (0x1C) – This is the value 1 (it equates to switch 29)
SV29 – 16 (0x10) – This is the value 2
What this will cause is that when the switch is thrown to the direction that +5 volts goes to JP1 pin 17 the following message is sent to the Loconet:
B01C10CC (all values are hex)
And the turnout is thrown to the thrown position
When the switch is thrown to the direction that 0 volts goes to JP1 pin 17 the following message is sent to the Loconet:
B01C30CC (all these values are hex)
And the turnout is thrown to the closed position

**Example 2**
Push button switch
The DT100 type throttle is put in switch mode and then turnout 29 is selected. Then the closed button is pressed and turnout 29 is closed. To have this turnout work off of a switch connected to LocoIO you would put the following values in the proper SV's. Lets assume that the switch is connected to pin 17 on the JP1 connector of LocoIO. Looking at the SV chart we see that SV's 27 – 29 apply to JP1 pin 17.
SV27 – 47 (0x2F) – This is the configuration SV for JP17
SV28 – 28 (0x1C) – This is the value 1 (it equates to switch 29)
SV29 – 16 (0x10) – This is the value 2
What this will cause is that when the push button switch is pressed and released the following message is sent to the Loconet:
B01C10CC (all values are hex)
And the turnout is thrown to the thrown position
When the push button switch is pressed and released the second time the following message is sent to the Loconet:
B01C30CC (all these values are hex)
And the turnout is thrown to the closed position
**Note:** This assumes that pressing the push button connects 0 volts to JP1 Pin 17 and releasing it connects +5 volts to JP1 pin 17.

**Note:** Using the value of 111 (0x6F) in the configuration byte will reverse the action (so pressing the button sends the message instead of releasing the button).

**To generate an OPC_INPUT_REP (0xB2) message (usually from a block detector)**

Block Detector
First you have to determine if the block detector is active high or active low.  If the block detectors are active high then you must set the setup SV (number 0) to have the #2 bit turned off (a zero in this field will work).  If the block detectors are active low then you must set the setup SV (number 0) to have the #2 bit turned on (a 04 in this field will work).
First you have to decide what block number you want to send out to the Loconet when a block becomes occupied (our example assumes block 04).  You then have to add 128 to the block number (turn the high order bit on).  Once you determine this then put the proper values in the proper SV's.  Lets assume that the block detector is connected to pin 17 on the JP1 connector of LocoIO.  Looking at the SV chart we see that SV's 27 – 29 apply to JP1 pin 17.
If SV0 bit2 = 0 (SV0 could = 0) then
SV27 – 15 (0x0F) – This is the configuration SV for JP17
SV28 – 132 (0x84) – This is the value 1 (this is for block 4 and the high order bit is turned on)
SV29 – 0 (0x00) – This is the value 2 (this can be set to 0x00 or 0x20 or 0x40 or 0x60)
What this will cause is that when the block is occupied the following message is sent to the Loconet:
B20410CC or B20430CC or B20450CC or B20470CC (all values are hex)
When the becomes unoccupied the following message is sent to the Loconet:
B20400CC or B20420CC or B20440CC or B20460CC (all these values are hex)

If SV0 bit2 = 1 (SV0 could = 4) then
SV27 – 15 (0x0F) – This is the configuration SV for JP17
SV28 – 132 (0x84) – This is the value 1 (this is for block 4 and the high order bit is turned on)
SV29 – 16 (0x10) – This is the value 2 (this can be set to 0x10 or 0x30 or 0x50 or 0x70)
What this will cause is that when the block is occupied the following message is sent to the Loconet:
B20410CC or B20430CC or B20450CC or B20470CC (all values are hex)
When the becomes unoccupied the following message is sent to the Loconet:
B20400CC or B20420CC or B20440CC or B20460CC (all these values are hex)

**To receive an OPC_INPUT_REP (0xB2) message (usually turns on a LED to indicate block occupancy)**

To program a SV to be an output SV you will do the following.

The input configuration value is 192 (0xC0).

When a OPC_INPUT_REP message is received a LED that is connected to an output will be turned on or off.  First decide what is going to generate the OPC_INPUT_REP message.  One way this is done is by a block detector.  To determine the value to put into Value 1 and Value 2 SV's you will have to do 1 of 2 things.  Read the Loconet Personnel Edition 1.0 and figure out what message will be generated or do a little test.  To do the test you will need something that shows Loconet packets.  This can be a MS100 and John Kabat's Lnmon software or any of the commercial products out there.  You can also use LocoBuffer and the RRCntl program or the DecoderPro program.  Start up the packet trace and then activate the device that will generate the OPC_INPUT_REP message.  If you are using a block detector to generate the message then just put an engine on the track and look at the 0xB2 message that is generated.  It will be something like (all these values are hex)
B20430CC or
B20420CC or
You will then use the 2nd byte for the Value 1 and the third byte (a 32 (0x20)) for Value 2.

B20410CC or
B20400CC
You will then use the 2nd byte for the Value 1 and the third byte (a 0) for Value 2.

**Example**
To have this message turn on a LED connected to LocoIO you would put the following values in the proper SV's. Lets assume that the LED is connected to pin 9 on the JP1 connector of LocoIO. Looking at the SV chart we see that SV's 15 – 17 apply to JP1 pin 9.
SV15 – 192 (0xC0) – This is the configuration SV for JP1 pin 9
SV16 – 04 (0x04) – This is the value 1
SV17 – 32 (0x20) – This is the value 2
What this will cause is that when the OPC_INPUT_REP message is received the LED connected to JP1 pin 9 will light up if a B20410CC message comes in and shut off if a B20400CC message comes in.

**To receive an OPC_SW_REQ (0xB0) message (usually to turn on a LED to indicate switch position)**

To program a SV to be an output SV you will do the following.

The input configuration value is 128 (0x80).

When a OPC_SW_REQ message is received a LED that is connected to an output will be turned on or off. First decide what is going to generate the OPC_SW_REQ message. One way this is done is by a switch or a throttle activating a turnout. To determine the value to put into Value 1 and Value 2 SV's you will have to do 1 of 2 things. Read the Loconet Personnel Edition 1.0 and figure out what message will be generated or do a little test. To do the test you will need something that shows Loconet packets. This can be a MS100 and John Kabat's Lnmon software or any of the commercial products out there. You can also use LocoBuffer and the RRCntl program or the DecoderPro program. Start up the packet trace and then activate the device that will generate the OPC_SW_REQ message. If you are using a throttle to generate the message then just select the turnout you are controlling and look at the 0xB0 message that is generated. It will be something like (all these values are hex)
B01C30CC or
B01C20CC or
You will then use the 2nd byte for the Value 1 and a 48 (0x30) for Value 2.

B01C10CC or
B01C00CC
You will then use the 2nd byte for the Value 1 and a 16 (0x10) for Value 2.

**Example**
To have this message turn on 2 different LEDs connected to LocoIO you would put the following values in the proper SV's. Lets assume that the LEDs are connected to pin 13 and pin 15 on the JP1 connector of LocoIO. Looking at the SV chart we see that SV's 21 – 26 apply to JP1 pin 13 and 15.
SV21 – 128 (0x80) – This is the configuration SV for JP1 pin 13
SV22 – 28 (0x1C) – This is the value 1
SV23 – 48 (0x30) – This is the value 2
SV24 – 128 (0x80) – This is the configuration SV for JP1 pin 15
SV25 – 28 (0x1C) – This is the value 1
SV26 – 16 (0x10) – This is the value 2
What this will cause is that when the OPC_SW_REQ message is received the LED connected to JP1 pin 13 will light up and the LED connected to JP1 pin 15 will shut off or vise versa.

**Multi port reads and writes**

The following commands are supported using the OPC_PEER_XFER message

03 – multi port write
04 – multi port read

This is useful for developing signaling systems. It allows you to change some or all LocoIO outputs at the same time. It also allows you to query the LocoIO to see how the ports are configured and their output state. The command uses the concept of a low data with a low data mask and a high data with a high data mask.

If you issue the 04 command in RRCntl to do a multi port read you are returned the following:

D5 – has the low 8 port direction
D6 – has the low 8 port values
D7 – has the high 8 port direction
D8 – has the high 8 port values

Where the low 8 ports are JP1 thru JP15 (JP1 is the most significant bit and JP 15 the least most significant bit). The high 8 ports are JP17 thru JP31 (JP17 is the most significant bit and JP 31 the least most significant bit).

Where direction is output if the bit is a 1 and input if the bit is 0.

If you want to change all or some of the ports at the same time you would use the 03 multi port write command. You would first read the LocoIO with the 04 multi port read command and then use the low and high mask you retrieved in D5 and D7 (remember where the high bits are if your not using RRCntl, in the PXCT2 field). This will tell you which bits are output and which bits are input. Using this data you then put the mask bits on the write command and the values you want to write in the low and high data fields and write the message. So D5 and D7 are port direction bytes in the 04 read command and they are data mask bytes in the 03 write message. If you don't understand this then maybe an example is in order.
If you issued a 04 multi port read command in RRCntl on a LocoIO that was just programmed by me, you would get the following values returned:
D5 255 0xFF
D6 0
D7 255 0xFF
D8 0
This says all 16 ports are output ports and they are all set to 0 volts.

Now if you issued a 03 multi port write command with the following values
D5 170 0xAA
D6 255 0xFF
D7 170 0xAA
D8 255 0xFF
This would set every other port to +5 volts (remember the mask only changes ports that have a 1 in the mask bit). You would get the same results if D6 and D8 were set to 170 (0xAA).

Another example is at the traffic signal section, which is off of the LocoIO page at
www.locobuffer.com and there is an example.

**Revision History**
Version 1 - original release
Version 1.1 – Push button support
Version 1.2 – Corrected a programming error problem
Version 1.3 – Added OPC_INPUT_REP (0xB2) output message option
Version 1.3.4 – Added LocoIO setup byte
Version 1.3.5 – Enhanced setup SV to allow block detectors to be active high or low

**PIC programming**

Version 1.3.5 of the PIC software is currently available.  The PIC in the device can be re-programmed using the ISP program, which is available at the LocoBuffer website www.locobuffer.com and a LocoBuffer device.  Optionally, you can design and build a programmer daughter board for LocoIO. You can also use a conventional PIC programmer or send me the PIC and I will program it for free.

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 ISP (In Circuit Serial Programming) – this requires a LocoBuffer.  This is quite easy to do and you can download ISP and read the ISP readme and then follow the on screen instructions.

**PC boards and kits**

I sell PC boards, programmed PIC's and kits.  Like I did with LocoBuffer, these will be sold at cost.  Due to the overwhelming request for LocoIO kits, I will be offering the kits using 25 unit pricing.  I have posted the LocoIO parts list (with current Digikey prices) in the files section of this list.
Complete "A" kits cost $23.00 and "B" kits go for $21.00.
Postage for 1 - 4 kits will be $3.85 using priority mail.
If you pay with PayPal, add 2.9% plus .30 cents (because that's what they charge me).
Kits include all the parts, PCB board and a programmed PIC.  They don't include a case or a wallwart power supply.
PCB boards are available separately for $7.30 plus $1.50 shipping.
PCB Board and a programmed PIC are available $12.00 plus $1.50 shipping.
All prices are in US dollars. Priority mail is only available to the US.
Shipping outside the US is different (and if you bought a LocoBuffer then you know what it is).

An "A" kit is all the parts listed in the parts list. A "B" kit is missing the voltage regulation parts and would get connected to the 5 volts supply of the "A" unit. The advantage of using the "A" and one or more "B" is less wallwart power supplies.

**Functionality**

The functionality of LocoIO right now is that it supports message OPC_SW_REQ (0xB0) and OPC_INPUT_REP (0xB2) going out and message OPC_INPUT_REP (0xB2) and OPC_SW_REQ (0xB0) coming in.

Example
You have an oval of track with 4 blocks, 4 turnouts and 4 sidings.  One LocoIO has 16 ports.  They could be configured as follows:

CTC1
port 1 - Block 12 works LED1
port 2 - Block 16 works LED2
port 3 - Switch 1 to control turnout 1
port 4 - Block 22 works LED3
port 5 - Switch 2 to control turnout 33
port 6 - Block 2 works LED4
port 7 - Switch 2 also controls LED15 (for turnout indication)
and so on.

CTC2
port 1 - LED 15
port 2 - Block 22 to control a signaling LED
and so on.

This assumes you have a block detector that sends out Loconet messages (such as a BD8 or BLD16) and that you have a stationary decoder to control the turnouts (such as a DS54 or AD4).

**Future**

I plan to add the following features.  If you have some that you would like, just ask and I will consider them.
Programming using DCC packets
Respond to a OPC_SW_STATE (0xBC) to tell the position of a switch
Respond to a GP_ON command to tell the position of a switch
Add turntable control
Add routes by using virtual ports

I can be reached at jjabour@optonline.net

**LocoIO Assembly Instructions**

First identify all the parts.

**CAUTION** – The PIC 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.

Power supplies

The LocoIO kit does not come with a power supply. You must supply one. If you are building an "A" kit then the power requirements are 7 – 15 volts AC or DC. Depending on the power requirements of the daughter board, current capacity should be between 100 – 1000 ma (please note the power limitations of the 16F873 PIC which are referenced else where in this document). This power gets connected to pads 1 and 2. Orientation is not important. If you are building a "B" kit then 5 volt regulates DC is a requirement. This gets connected to pads 3 and 4 and orientation is **very** important. The pads are marked with a + (for +5 volts) and a – (for ground).

All statements relating to orientation are with the top of the board up and the word LocoIO 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

NOTE: Do not install R11 at this time. This is for a future upgrade and may effect the operation of LocoIO. Also R10 does not exist.

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.

Note: Pin 1 on the 2 IC's are pointing in different directions.

14 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.
Note: The "B" kit will not have this part.

JP1 36 pin header

JP2 is not used at this time but install it now and add the jumper to it in either position so that the PIC has a known state on that pin.

J1 – RJ12 6 pin Loconet connector

J2 – 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.

IC3 – 7805 voltage regulator – large black rectangular device with 3 heavy leads coming out the bottom. Orientation is important. The words must point to the left. If this is backwards you won't get +5volts out of the device. This is the opposite direction of the LocoBuffer.
Note: The "B" kit will not have this part.

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 LocoIO. If this is backwards you won't get +5volts out of the device.
Note: The "B" kit will not have this part.

D1 – Red LED – Insert with the long lead to the top, short lead to the word LocoIO. When power is applied, this should light.

D2 – Red LED – Insert with the long lead to the top, short lead to the word LocoIO. This will flash with Loconet packets

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 to the right. 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.

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

Congratulations. You have built a LocoIO mainboard. Now lets see if it works. To do this you need a program that will put out a OPC_PEER_XFER message. The sample program RRCntl will do this. It requires a LocoBuffer to run.
Note: Do not hook up any devices to the JP1 header at this time.

**Testing**

Connect a straight through active Loconet cable to the LocoIO. Plug in the wallwart. Startup the sample RRCntl program on your PC. Set the serial port to the correct baud and to the correct Comm port. Then open the port. Now open the debug window and the LocoIO window (from the Tools menu). Click on the send key and you should see the following 2 OPC_PEER_XFER messages:

E5105051010002010000100000000019
E5105150010000000000010000000514B

If you do then you were successful. If you don't, then try a few more times. If you get the 2 messages on the second or third try then LocoIO is working. If not then go to the troubleshooting section of the LocoIO documentation.

A few more tests you can do to verify that the device works and to get familiar with the functionality of the device is to do the following.
All the even numbered pins (from 2 – 34) on JP1 are ground pins at 0 volts.
Even numbered pins are near the PCB board edge.
Pin 1 of JP1 is the pin closest to the RJ12 connectors.
Pins 35 and 36 are +5 volts.
All the odd number pins (from 1 to 33) on JP1 should be at 0 volts

Send the following message out to the Loconet
B201105C
Pin 1 should have changed from 0 volts to +5 volts
Send the following message out to the Loconet
B201004C
Pin 1 should have changed from +5 volts to 0 volts
No other pins should have changed their state
You can do this test on the first 8 ports using 1 – 4 for the second byte and alternating the third byte between 16 (0x10) and 0 and between 32 (0x20) and 48 (0x30)

Send the following message out to the Loconet
B001105E
Pin 17 should have changed from 0 volts to +5 volts and pin 19 should remain at 0 volts
Send the following message out to the Loconet
B001307E
Pin 17 should have changed from +5 volts to 0 volts and pin 19 should have changed to +5 volts
No other pins should have changed their state
You can do this test on the second 8 ports using 1 – 4 for the second byte and alternating the third byte between 16 (0x10) and 48 (0x30)

The last test is to program SV 3 to a value of 15 (0x0F). Then cycle power to the LocoIO device.
Now, carefully bring pin 1 of JP1 to +5 volts. The following packet will show up on the Loconet

B001307E
Now, carefully bring pin 1 of JP1 to 0 volts.   The following packet will show up on the Loconet
B001105E

The next thing you must do before you connect anything to the JP1 header is to set the SV's to values that match the devices that you are going to connect to JP1.  All ports are defaulting to output ports.  You must configure any ports that are input ports at this time.  Once the ports are configured to output or input then power to the LocoIO should be cycled and then the devices can be connected to JP1 on the LocoIO. Failure to do this may cause unpredictable results and the Loconet getting flooded with messages.  It probably won't burn out any of the IC's but good electrical practices should always be followed. Some of these are:
Don't leave PIC inputs in a state where they can float (they should always be at 0 or +5 volts)
Watch power consumption.  The Microchip data sheet defines the power capabilities of the 16F873 PIC processor and should be consulted for the actual values (but roughly they say that each pin can source and/ or sink 25ma and that all the ports on PortA and PortB can not exceed 200ma).

**Troubleshooting 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)

If there are any soldering problems, they will probably be on the JP1 header.

Transistor problems
The transistor I used is a 2N4401 because it works.  Other people have used a BC337 or 2N2222.
Any transistor (npn I believe) with enough gain should work.
If this is put in the circuit wrong the output won't work.

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

PIC Pin 2 (Loconet out from LocoIO)
should be low until the LocoIO sends a Loconet packet
out and then it should flash.

Pin 3 (Loconet in from Loconet)
should basically be high and flash whenever Loconet
packets are present.
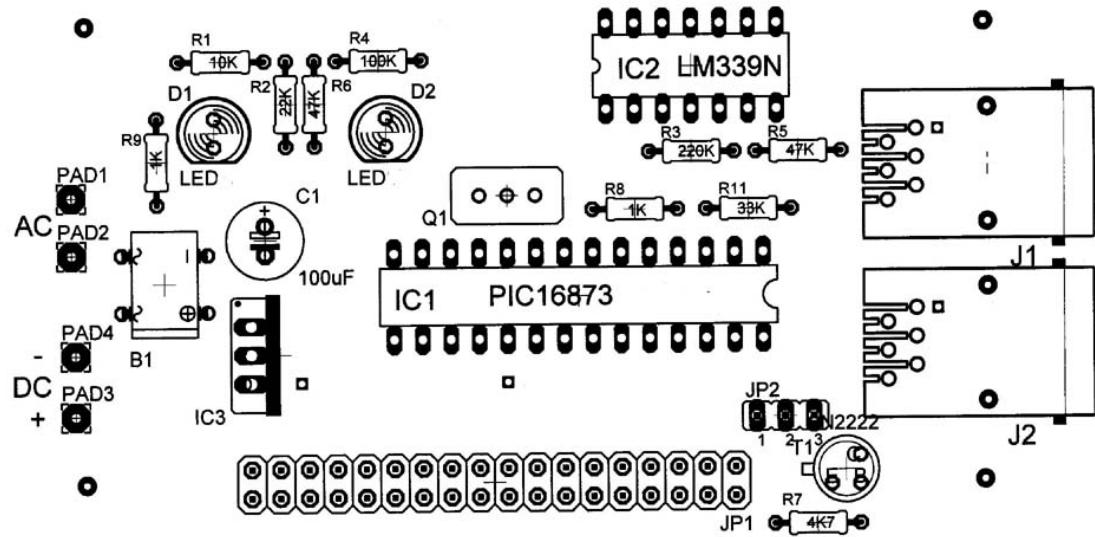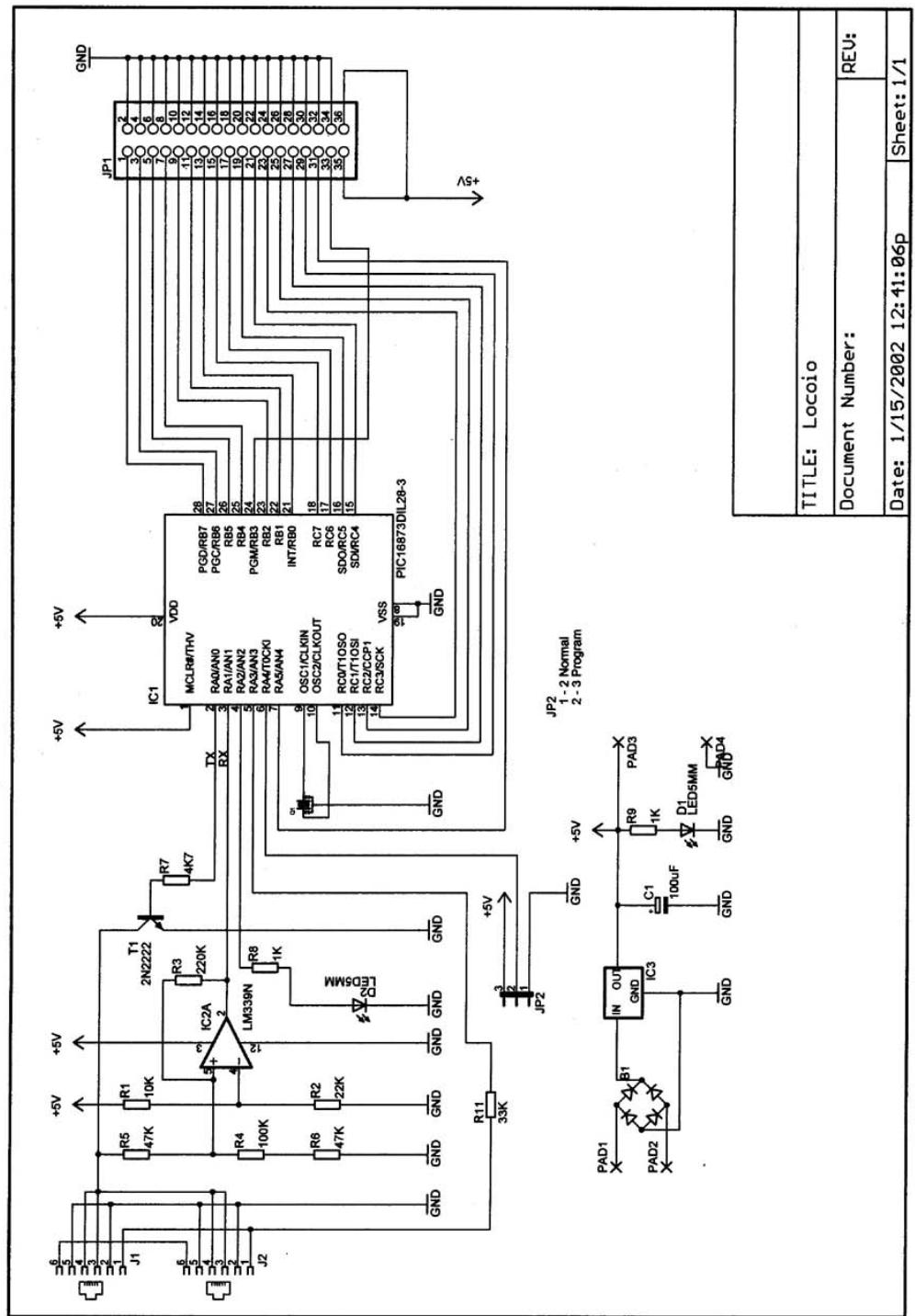
R1
10K

R4
100K

R2
22K

R6
47K

D1

D2

R9
1K

LED

LED

PAD1

AC

PAD2

C1

100uF

IC2 LM339N

R3
220K

R5
47K

R8
1K

R11
33K

Q1

PAD4

−

B1

DC

PAD3

+

IC3

IC1    PIC16873

JP2

1    2  3

2N2222

T1

R7
4K7

JP1

J1

J2

TITLE: Locoio

Document Number:

REV:

Date: 1/15/2002 12:41:06p    Sheet: 1/1

| Socket/Part | Description/Label/IO Setup | JP1 # / Vendor | PIC Pin / Digikey | PIC name / Price | Function / control bits Setup | Default / 0 | Hex / 0 | Comment / Quantity |
|---|---|---|---|---|---|---|---|---|
| PIC16F872-20/SP-ND Device Low Address | Microchip | | | $7.63 | I can program this for free | 81 | 51 | 1 |
| X902-ND Device High Address | | | | $0.81 | Ceramic Resonator w/cap 20mhz | 1 | 20mhz | 1 |
| ED3328-ND Configuration | | 1 | 28 | RB7 $1.46 | 28 pin DIP socket .300 | 192 | C0 | 1 |
| A9414-ND Value 1 | | | | $0.75 | 14 pin DIP socket | 1 | 1 | 1 |
| 296-5303-5-ND | | | | $0.52 | LM339N Quad comparator | 0 | 0 | 1 |
| NJM7805FA-ND Configuration | | 3 | 27 | RB6 $0.57 | Voltage regulator | 192 | C0 | 1 |
| A9049-ND Value 1 | | | | $1.44 | 6-Conn connector for RJ12 | 1 | 1 | 2 |
| DB101-ND Value 2 | | | | $0.52 | Bridge rectifier | 32 | 20 | 1 |
| 2N4401 Configuration | | 5 | 26 | RB5 $0.21 | 2N4401 Transistor | 192 | C0 | 1 |
| AP Value 1 | | | | $7.30 | PC board | 2 | 2 | 1 |
| 929834-02-36-ND | | | | $0.20 | Header | 0 | 0 | 0.2 |
| 929836-00-36-ND Configuration | | 7 | 25 | RB4 $1.01 | Header | 192 | C0 | 0.5 |
| 929950-00-ND | | | | $0.15 | jumpers | 2 | 2 | 1 |
| 1.0KQBK-ND Value 2 | | | | $0.12 | 1K Value2 | 32 | 20 | 2 |
| 47KQBK-ND Configuration | | 9 | 23 | RB0 $0.12 | 47K Configuration | 192 | C0 | 2 |
| 100KQBK-ND Value 1 | | | | $0.06 | 100K Value1 | 3 | 3 | 1 |
| 33KQBK-ND Value 2 | | | | $0.06 | 33K Value2 | 0 | 0 | 1 |
| 220KQBK-ND Configuration | | 11 | 22 | RB1 $0.06 | 220K Configuration | 192 | C0 | 1 |
| 22KQBK-ND Value 1 | | | | $0.06 | 22K Value1 | 3 | 3 | 1 |
| 10KQBK-ND Value 2 | | | | $0.06 | 10K Value2 | 32 | 20 | 1 |
| 4.7KQBK-ND Configuration | | 13 | 21 | RB2 $0.06 | 4.7K Configuration | 192 | C0 | 1 |
| P10229-ND Value 1 | | | | $0.63 | 100uf Value1 | 4 | 4 | 1 |
| P363-ND Value 2 | | | | $0.34 | Red LED Value2 | 0 | 0 | 2 |
| 24 Configuration | | 15 | 18 | RC7 | Configuration | 192 | C0 | |
| 25 Value 1 | | | | | Value1 | 4 | 4 | |
| 26 Value 2 (Total) | | | | $24.14 | Value2 | 32 | 20 | |
| 27 Configuration | | 17 | 17 | RC6 | Configuration | 128 | 80 | |
| 28 Value 1 | | | | | Value1 | 1 | 1 | |
| 29 Value 2 (Digikey prices are in units of one as of 01/14/02) | | | | | Value2 | 16 | 10 | |
| 30 Configuration (Power Supply required (7-15 volts AC or DC wall wart)) | | 19 | 16 | RC5 | Configuration | 128 | 80 | |
| 31 Value 1 (Resistor wattage is not important (1/4 watt 5% fits board)) | | | | | Value1 | 1 | 1 | |
| 32 Value 2 | | | | | Value2 | 48 | 30 | |
| 33 Configuration | | 21 | 15 | RC4 | Configuration | 128 | 80 | |
| 34 Value 1 | | | | | Value1 | 2 | 2 | |
| 35 Value 2 | | | | | Value2 | 16 | 10 | |
| 36 Configuration | | 23 | 14 | RC3 | Configuration | 128 | 80 | |
| 37 Value 1 | | | | | Value1 | 2 | 2 | |
| 38 Value 2 | | | | | Value2 | 48 | 30 | |
| 39 Configuration | | 25 | 13 | RC2 | Configuration | 128 | 80 | |
| 40 Value 1 | | | | | Value1 | 3 | 3 | |
| 41 Value 2 | | | | | Value2 | 16 | 10 | |
| 42 Configuration | | 27 | 12 | RC1 | Configuration | 128 | 80 | |
| 43 Value 1 | | | | | Value1 | 3 | 3 | |
| 44 Value 2 | | | | | Value2 | 48 | 30 | |
| 45 Configuration | | 29 | 11 | RC0 | Configuration | 128 | 80 | |
| 46 Value 1 | | | | | Value1 | 4 | 4 | |
| 47 Value 2 | | | | | Value2 | 16 | 10 | |
| 48 Configuration | | 31 | 7 | RA5 | Configuration | 128 | 80 | |
| 49 Value 1 | | | | | Value1 | 4 | 4 | |
| 50 Value 2 | | | | | Value2 | 48 | 30 | |
| 51 Configuration | | 33 | 24 | RB3 | | | | |
| 52 Value 1 | | | | | | | | |
| 53 Value 2 | | | | | | | | |
| Reserved 54 - 127 | | | | | | | | |