NeoLoch

Inquisitor 6821 PIA Blade Manual

Overview

The Inquisitor PIA blade is designed to test 6821, and compatible, Peripheral Interface Adapters. The Inquisitor PIA tester conducts a wide range of tests to discover problems with a PIA. These problems are then accessible via the up and down arrow keys.

Preliminary Release

This document details the operation of the default configuration of the IC tester as well as details on the device's operation for custom code design.

Table of Contents

1.0 Device Connection

1.1 CN1 – Card Edge Connector

Edge of PC board that plugs into the card edge socket on the Inquisitor Core PC board. When plugging in the SRAM Blade, line up the white triangle with the white triangle on the Core PC board.

1.2 CN2 – ICSP (In-Circuit Serial Programming)

ICSP connector, this port is designed to attach to a PICkit 3 or compatible programmer.

1.3 LED 1 – Power LED

Indicates when power is applied to the Blade.

2.0 Understanding PIA Tester Process

To begin testing a PIA, use the following steps:

- Insert the PIA blade into the Core module.
- Power up the tester.
- Insert the PIA into the ZIF socket, make sure pin 1 is the closest pin to the ZIF's lever.
- Lower the lever to lock the PIA in place.
- Press the left or right arrow key to select 6821 test.
- Press the enter key to start the test.

The Inquisitor PIA tester has a total of 29 tests that the PIA will be subjected to. Each test is conducted at least 100 times, though for PORTA/B read / write operations this number is 256. Follows is a detailed description of each test and it's associated error number.

2.1.E0 – PORTA Read

Test the PORTA output function by configuring this port as output and then writing values from 0x00 to 0xFF. This process consists of:

- Configure PORTA as output.
- Data is then written to the port.
- Data is read from the port by the MCU and compared. If at any point the data doesn't match, the test fails.

When displaying this error on the LCD screen the bad bits will appear on the second line.

LCD Display L1: E00:PORTA Read
L2: x x x x x x x x

Data Flow:	MCU	→ PIA	→ Port A	→ MCU

2.1.E1 – PORTA Write

Test the PORTA input function by configuring this port as input and then writing values from 0x00 to 0xFF. This process consists of:

- Configure PORTA as input.
- Data is then written to the port.
- Data is read from the PIA by the MCU and compared. If at any point the data doesn't match, the test fails.

When displaying this error on the LCD screen the bad bits will appear on the second line.

LCD Display L1: E01:PORTA Write
L2: x x x x x x x x

Data Flow:	MCU	→ Port A	→ PIA	→ MCU

2.1.E2 – PORTB Read

Functionally the same as PORTA read.

LCD Display L1: E02:PORTB Read
L2: x x x x x x x x

Data Flow:	MCU	→ PIB	→ Port B	→ MCU

2.1.E3 – PORTB write

Functionally the same as PORTA write.

LCD Display L1: E03:PORTB Write
L2: x x x x x x x x

Data Flow:	MCU	→ Port B	→ PIB	→ MCU

2.1.E4 & E5 – $\overline{\text{IRQA}}$ Using CA1 With Rising Edge

This test consists of testing $\overline{\text{IRQA}}$ and bit 7 of the PORTA control register. Errors for this test are reported as E4 ($\overline{\text{IRQA}}$) and E5 (Control Register b7). This test consists of:

- Configuring the PIA for CA1 interrupts with a rising edge.
- Triggering the interrupt on CA1.
- Checking $\overline{\text{IRQA}}$ to see if the interrupt was detected.
- Checking bit 7 of CONA for the interrupt flag.
- Reading PORTA to reset the flag and then checking bit 7 again for reset.

LCD Display L1: E04:CA1 P40 ↑E
L2: :IRQA P38

LCD Display L1: E05:PIA Read ↑E
L2: CONA Bit 7

Data Flow:	MCU	→ CA1	→ PIA	→ $\overline{\text{IRQA}}$	→ MCU
				→ Bit 7	→ MCU

2.1.E6 – CA2 Strobe With CA1 Restore – Rising Edge

CA2 is configured to strobe when data is read from PORTA, CA2 is restored via CA1.

LCD Display L1: E06:CA2 P39 Strb
L2: ↑E CA1-P25 Res.

Data Flow:	MCU	→ CA1	→ PIA	→ CA2	→ MCU

2.1.E7 & E8 – $\overline{\text{IRQA}}$ Using CA1 With Falling Edge

This test consists of testing $\overline{\text{IRQA}}$ and bit 7 of the PORTA control register. Errors for this test are reported as E7 ($\overline{\text{IRQA}}$) and E8 (Control Register b7). This test consists of:

- Configuring the PIA for CA1 interrupts with a falling edge.
- Triggering the interrupt on CA1.
- Checking $\overline{\text{IRQA}}$ to see if the interrupt was detected.
- Checking bit 7 of CONA for the interrupt flag.
- Reading PORTA to reset the flag and then checking bit 7 again for reset.

LCD Display L1: E07:CA1 P40 ↓E
L2: :IRQA P38

LCD Display L1: E08:PIA Read ↓E
L2: CONA Bit 7

Data Flow:	MCU	→ CA1	→ PIA	→ $\overline{\text{IRQA}}$	→ MCU
				→ Bit 7	→ MCU

2.1.E9 – CA2 Strobe With CA1 Restore – Falling Edge

CA2 is configured to strobe when data is read from PORTA, CA2 is restored via CA1.

LCD Display L1: E09:CA2 P39 Strb
L2: ↓E CA1-P25 Res.

Data Flow:	MCU	→ CA1	→ PIA	→ CA2	→ MCU

2.1.E10 – CA2 Strobe With E Restore

CA2 is configured to strobe when data is read from PORTA, CA2 is restored via E.

LCD Display L1: E09:CA2 P39 Strb
L2:With E P25 Res.

Data Flow:	MCU	→ E	→ PIA	→ CA2	→ MCU

2.1.E11 – CA2 Output Mode

CA2 is configured to act as an output.

LCD Display L1: E09:CA2 P39
L2:Output Mode.

Data Flow:	MCU	→ CONA	→ CA2	→ MCU

2.1.E12 & E13 – $\overline{\text{IRQA}}$ Using CA2 With Falling Edge

This test consists of testing $\overline{\text{IRQA}}$ and bit 6 of the PORTA control register. Errors for this test are reported as E12 ($\overline{\text{IRQA}}$) and E13 (Control Register b6). This test consists of:

- Configuring the PIA for CA2 interrupts with a falling edge.
- Triggering the interrupt on CA2.
- Checking $\overline{\text{IRQA}}$ to see if the interrupt was detected.
- Checking bit 6 of CONA for the interrupt flag.
- Reading PORTA to reset the flag and then checking bit 6 again for reset.

LCD Display L1: E12:CA2 P39 ↓E
L2: :IRQA P38

LCD Display L1: E13:PIA Read ↓E
L2: CONA Bit 6

Data Flow:	MCU	→ CA2	→ PIA	→ $\overline{\text{IRQA}}$	→ MCU
				→ Bit 6	→ MCU

2.1.E14 & E15 – $\overline{\text{IRQA}}$ Using CA2 With Rising Edge

This test consists of testing $\overline{\text{IRQA}}$ and bit 6 of the PORTA control register. Errors for this test are reported as E12 ($\overline{\text{IRQA}}$) and E13 (Control Register b6). This test consists of:

- Configuring the PIA for CA2 interrupts with a rising edge.
- Triggering the interrupt on CA2.
- Checking $\overline{\text{IRQA}}$ to see if the interrupt was detected.
- Checking bit 6 of CONA for the interrupt flag.
- Reading PORTA to reset the flag and then checking bit 6 again for reset.

LCD Display L1: E14:CA2 P39 ↑E
L2: :IRQA P38

LCD Display L1: E15:PIA Read ↑E
L2: CONA Bit 6

Data Flow:	MCU	→ CA2	→ PIA	→ $\overline{\text{IRQA}}$	→ MCU
				→ Bit 6	→ MCU

2.1.E16 & E17 – $\overline{\text{IRQB}}$ Using CB1 With Falling Edge

This test consists of testing $\overline{\text{IRQB}}$ and bit 7 of the PORTB control register. Errors for this test are reported as E16 ($\overline{\text{IRQB}}$) and E17 (Control Register b7). This test consists of:

- Configuring the PIA for CB1 interrupts with a falling edge.
- Triggering the interrupt on CB1.
- Checking $\overline{\text{IRQB}}$ to see if the interrupt was detected.
- Checking bit 7 of CONB for the interrupt flag.
- Reading PORTB to reset the flag and then checking bit 7 again for reset.

LCD Display L1: E16:CB1 P18 ↓E
L2: :IRQB P37

LCD Display L1: E17:PIB Read ↓E
L2: CONB Bit 7

Data Flow:	MCU	→ CB1	→ PIA	→ $\overline{\text{IRQB}}$	→ MCU
				→ Bit 7	→ MCU

2.1.E18 & E19 – $\overline{\text{IRQB}}$ Using CB1 With Rising Edge

This test consists of testing $\overline{\text{IRQB}}$ and bit 7 of the PORTB control register. Errors for this test are reported as E18 ($\overline{\text{IRQB}}$) and E19 (Control Register b7). This test consists of:

- Configuring the PIA for CB1 interrupts with a rising edge.
- Triggering the interrupt on CB1.
- Checking $\overline{\text{IRQB}}$ to see if the interrupt was detected.
- Checking bit 7 of CONB for the interrupt flag.
- Reading PORTB to reset the flag and then checking bit 7 again for reset.

LCD Display L1: E18:CB1 P18 ↑E
L2: :IRQB P37

LCD Display L1: E19:PIB Read ↑E
L2: CONB Bit 7

Data Flow:	MCU	→ CB1	→ PIA	→ $\overline{\text{IRQB}}$	→ MCU
				→ Bit 7	→ MCU

2.1.E20 & E21 – $\overline{\text{IRQB}}$ Using CB2 With Falling Edge

This test consists of testing $\overline{\text{IRQB}}$ and bit 6 of the PORTB control register. Errors for this test are reported as E20 ($\overline{\text{IRQB}}$) and E21 (Control Register b6). This test consists of:

- Configuring the PIA for CB2 interrupts with a falling edge.
- Triggering the interrupt on CB2.
- Checking $\overline{\text{IRQB}}$ to see if the interrupt was detected.
- Checking bit 6 of CONB for the interrupt flag.
- Reading PORTB to reset the flag and then checking bit 6 again for reset.

LCD Display L1: E20:CB2 P19 ↓E
L2: :IRQB P37

LCD Display L1: E21:PIB Read ↓E
L2: CONB Bit 6

Data Flow:	MCU	→ CB1	→ PIA	→ $\overline{\text{IRQB}}$	→ MCU
				→ Bit 6	→ MCU

2.1.E22 & E23 – $\overline{\text{IRQB}}$ Using CB2 With Rising Edge

This test consists of testing $\overline{\text{IRQB}}$ and bit 6 of the PORTB control register. Errors for this test are reported as E22 ($\overline{\text{IRQB}}$) and E23 (Control Register b6). This test consists of:

- Configuring the PIA for CB2 interrupts with a rising edge.
- Triggering the interrupt on CB2.
- Checking $\overline{\text{IRQB}}$ to see if the interrupt was detected.
- Checking bit 6 of CONB for the interrupt flag.
- Reading PORTB to reset the flag and then checking bit 6 again for reset.

LCD Display L1: E20:CB2 P19 ↑E
L2: :IRQB P37

LCD Display L1: E21:PIB Read ↑E
L2: CONB Bit 6

Data Flow:	MCU	→ CB1	→ PIA	→ $\overline{\text{IRQB}}$	→ MCU
				→ Bit 6	→ MCU

2.1.E24 – CB2 Strobe With CB1 Restore – Falling Edge

CB2 is configured to strobe when data is written to PORTB, CB2 is restored via CB1. This test consists of:

- Configure CB2 for strobe mode with CB1 restore.
- Write to PORTB.
- Bring E low, then high.
- Check CB2 for low state.
- Bring CB1 low, then high.
- Read PIB to clear interrupt.
- Check CB2 for high state.

LCD Display L1: E20:CB2 P19 Strb
L2:↓E CB1 P37 Res.

Data Flow:	MCU	→ PIB	→ E	CB2	→ MCU	→ CB1
	CB2	→ MCU				

2.1.E25 – CB2 Strobe With CB1 Restore – Rising Edge

CB2 is configured to strobe when data is written to PORTB, CB2 is restored via CB1.

- Configure CB2 for strobe mode with CB1 restore.
- Write to PORTB.
- Bring E low, then high.
- Check CB2 for low state.
- Bring CB1 low, then high.
- Read PIB to clear interrupt.
- Check CB2 for high state.

LCD Display L1: E20:CB2 P19 Strb
L2:↑E CB1 P37 Res.

Data Flow:	MCU	→ PIB	→ E	CB2	→ MCU	→ CB1
	CB2	→ MCU				

2.1.E26 – CB2 Strobe With E Restore

CB2 is configured to strobe when data is read from PORTA, CB2 is restored via E.

- Configure CB2 for strobe mode with E restore.
- Write to PORTB.
- Bring E low, then high.
- Check CB2 for low state.
- Deselect PIA
- Bring E low, then high.
- Check CB2 for high state.

LCD Display L1: E20:CB2 P19 Strb
L2:With E P25 Res.

Data Flow:	MCU	→ PIB	→ E	CB2	→ MCU	→ E
	CB2	→ MCU				

2.1.E27 – CB2 Output

CB2 is configured to strobe when data is read from PORTA, CB2 is restored via E.

- Configure CB2 for output mode.
- Write zero to CONB bit 3
- Check CB2 to see if it's low.
- Write one to CONB bit 3.
- Check CB2 to see if it's high.

LCD Display L1: E27:CB2 Output.
L2:

Data Flow:	MCU	→ CONB	→ MCU

2.1.E28 – Reset Cycle

The RESET line is cycled and then all internal registers are checked for proper reset values.

2.1.E29 – Excessive Current

This error means that the MCU has detected a voltage drop after powering up the PIA. This error may not appear when using different power sources, it all depends on how much current the power supply you are using can safely supply. Even if the PIA is pulling a lot of current, if there's no voltage drop on the system then this error will not be triggered.

Once an excessive current event is detected, the text "*HC" will appear on the right side of the second line on the LCD screen. The MCU then aborts testing the PIA and reports the PIA as bad. The system voltage is checked throughout the PIA testing process, so it's possible for the excessive current error to appear at any time.

However, the abort can be turned off by using the left or right arrow key to select the "High Current" flag and then changing the setting by using the enter key. This is only temporary and the PIA will default to the normal setting the next time it's powered up.

* IMPORTANT *
Running a power source that can easily supply 500ma or more or turning off the excessive current feature may cause damage to the tester. Be advised that the transistor used to control ground to the PIA has a limit of 600 ma. And the relay has a limit of 1A.

Symptoms of High Current Draw:

- LCD Screen dims.
- Transistor gets hot.
- +5V regulator gets hot.
- The PIA under test gets hot.

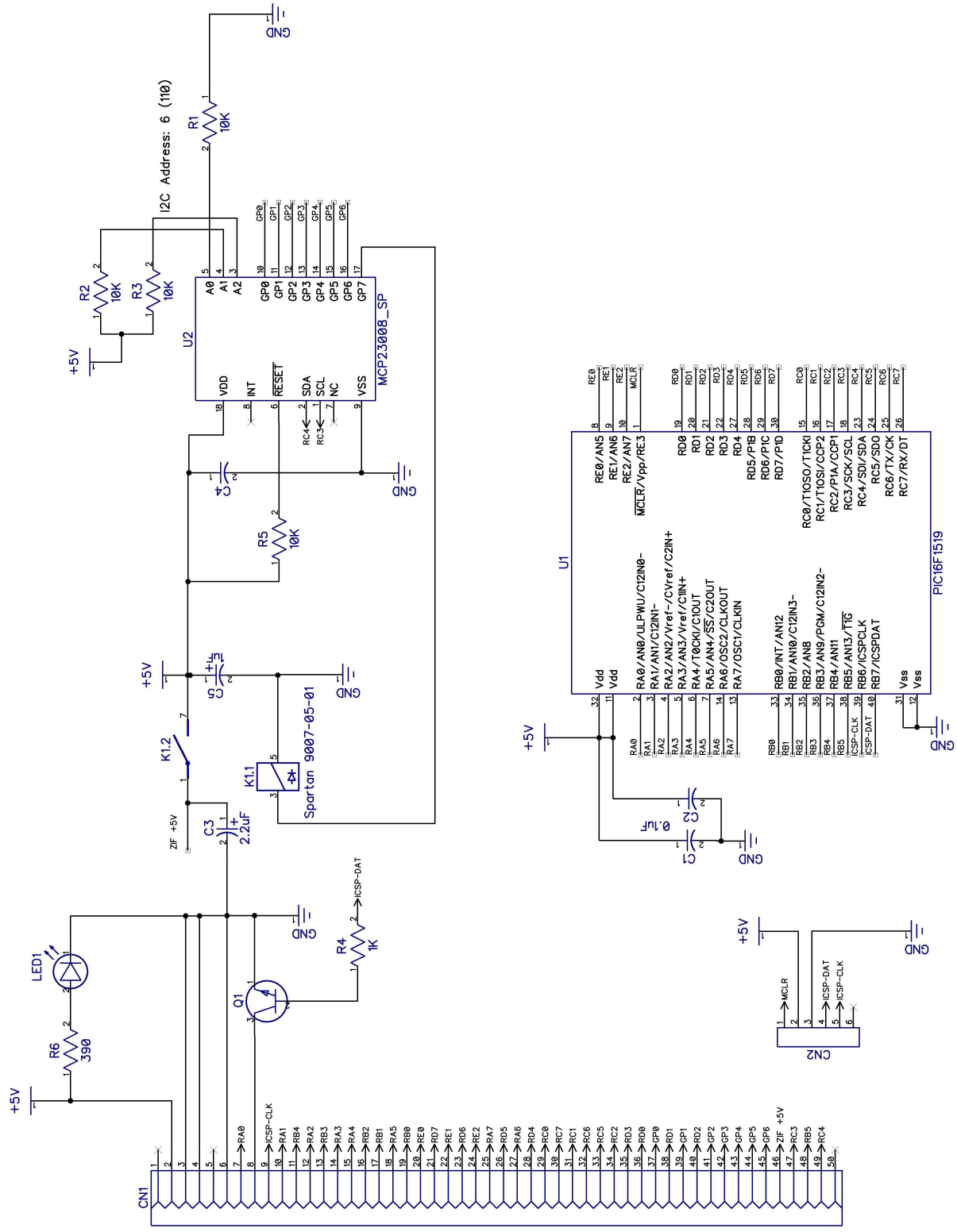
2.2 LED Result

At the conclusion of a test, LED 3 on the main board will visually display the result. Green means the PIA passed and red means the PIA failed. During the test the LED will be orange.

2.3 Display PIA Results.

If a PIA tests bad, the specific tests that failed will be available for viewing by using the up and down arrow keys on the main board. Only those tests that failed will be available for display.

3.0 Schematic



4.0 Card Edge to MCU / Port Expander Connections

Below is a table that lays out the pin assignment between the card edge connector, the MCU and the Port Expander.

Card Pin #	Function	Card Pin #	Function
2	+5V	1	+12V
4	Ground	3	Ground
6	Card Error LED (Gnd to turn LED off)	5	Not Connected
8	ZIF Pin 1 – Q1	7	ZIF Pin 40 – +5V via Relay
10	ZIF Pin 2 – MCU RA1	9	ZIF Pin 39 – MCU RB6
12	ZIF Pin 3 – MCU RA2	11	ZIF Pin 38 – MCU RB4
14	ZIF Pin 4 – MCU RA3	13	ZIF Pin 37 – MCU RB3
16	ZIF Pin 5 – MCU RB2	15	ZIF Pin 36 – MCU RA4
18	ZIF Pin 6 – MCU RA5	17	ZIF Pin 35 – MCU RB1
20	ZIF Pin 7 – MCU RE0	19	ZIF Pin 34 – MCU RB0
22	ZIF Pin 8 – MCU RE1	21	ZIF Pin 33 – MCU RD7
24	ZIF Pin 9 – MCU RE2	23	ZIF Pin 32 – MCU RD6
26	ZIF Pin 10 – MCU RD5	25	ZIF Pin 31 – MCU RA7
28	ZIF Pin 11 – MCU RD4	27	ZIF Pin 30 – MCU RA6
30	ZIF Pin 12 – MCU RC7	29	ZIF Pin 29 – MCU RC8
32	ZIF Pin 13 – MCU RC6	31	ZIF Pin 28 – MCU RC1
34	ZIF Pin 14 – MCU RC2	33	ZIF Pin 27 – MCU RC5
36	ZIF Pin 15 – MCU RD0	35	ZIF Pin 26 – MCU RD3
38	ZIF Pin 16 – MCU RD1	37	ZIF Pin 25 – Port Expander GP0
40	ZIF Pin 17 – MCU RD2	39	ZIF Pin 24 – Port Expander GP1
42	ZIF Pin 18 – Port Expander GP3	41	ZIF Pin 23 – Port Expander GP2
44	ZIF Pin 19 – Port Expander GP5	43	ZIF Pin 22 – Port Expander GP4
46	ZIF Pin 20 – Relay +5V	45	ZIF Pin 21 – Port Expander GP6
48	Switches – MCU RB5	47	I2C – SCL – MCU RC3
50	Not Connected	49	I2C - SDA – MCU RC4

5.0 Parts List

- 1 – Printed circuit board
- 1 – 40 pin socket
- 1 – 18 pin socket
- 1 – PIC16F1519 microcontroller
- 1 – MCP23008 port expander
- 1 – Reed Relays (Polarized)
- 4 – 10K Ohm Resistor (Brown Black Orange)
- 1 – 1K Ohm Resistor (Brown Black Red)
- 1 – 390 Ohm Resistor (Orange, White, Brown)
- 1 – 2N4401 NPN transistors
- 1 – 2x5 rectangle green LED
- 3 – 0.1uF Ceramic Capacitors
- 1 – 2.2uF capacitor
- 1 – 1.0 uF Electrolytic Capacitor

Appendix A: Firmware Revision History

Firmware Version 1.01

- Initial Release

Firmware Version 1.02

- Corrected an issue that in some rare cases (less than 1%), the MCU would power upon the "I Abort" setting instead of the 6821 test.

Appendix B: Document Revision History

Revision A (11/2015)

- Initial release of this document