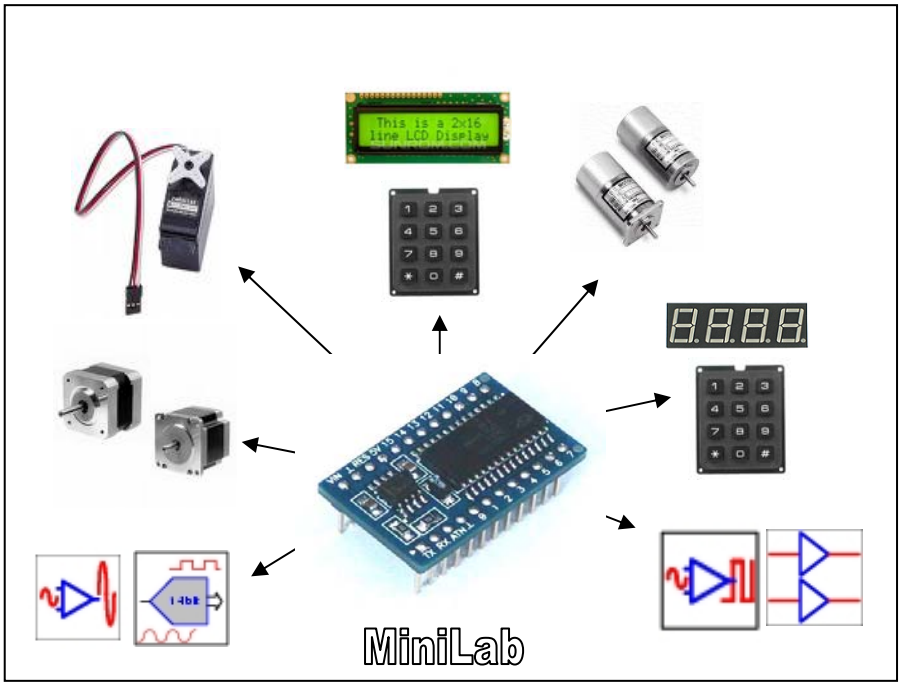
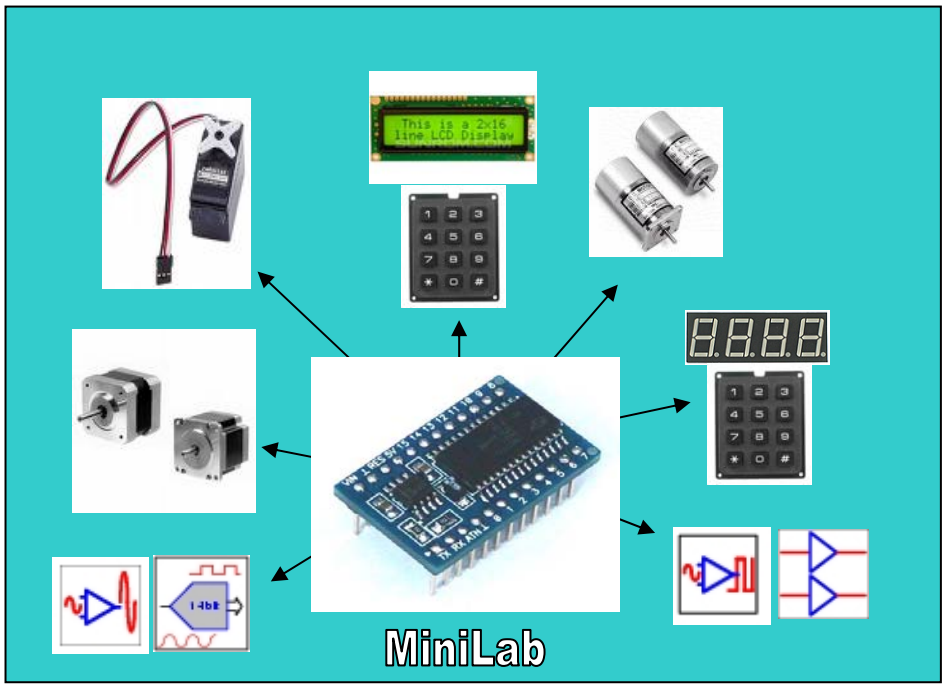


Preprogrammed dynamically reconfigurable device «MiniLab»

User manual

CONTENTS

1 Introduction.



MiniLab device includes configurable blocks of analog circuits and digital logic, as well as programmable interconnect. This architecture allows the user to create customized peripheral configurations, to match the requirements of each individual application.

Multiple different hardware function sets can be implemented in one single device under software control. It enables a designer to dynamically change the configurations repeatedly “on-the-fly” while the device is running.

MiniLab can be directly connected to yours microcontroller or computer as the intelligent peripherals (coprocessor).

The device supports an I2C and serial TTL interface and uses only two lines to communicate with a main controller (microcontroller or PC). The simple ASCII commands allow easy module control from microcontroller or PC.

Two MiniLab devices are available:

- MiniLab with I2C interface
- MiniLab with Serial TTL interface

Each device includes 8 preprogrammed configurations. The configurations can be switched in run time. The dynamic reconfiguration time is less than 200 µs.

MiniLab device includes the configurations as shown in Table 1.

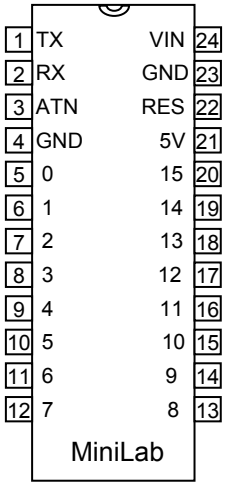
Table1.

Configuration Number	Description
1	16 channels RC servo control
2	Data Acquisition System: 4 analog channels with programmable gain amplifier, 14 bit ADC, two 9 bit analog outputs
3	Character LCD control (supports LCDs from 8x1 to 20x4)
4	LED 7-Segment Display control (supports Single-Digit, Dual-Digit, Triple-Digit, Quad-Digit Displays)
5	Up to 4 DC motor control (includes four 8 bit PWM blocks)
6	Stepper motor control
7	RGB LED control (control RGB LEDs with 24 bit color)
8	Analog Tool Set: 14 bit ADC, 9 bit analog output, two programmable gain amplifiers, comparator with programmable reference level

2 MINILAB DESCRIPTION.

2.1 MiniLab board and pin assignments.

The MiniLab device uses surface mount components to fit in a small 24-pin package. The device connector has 0.1” pin spacing and 0.6” pin row distance for easy prototyping and integration. The board includes a voltage regulator and a preprogrammed dynamically reconfigurable chip. Most the I/O Pins have different functionality depending on the module configuration.



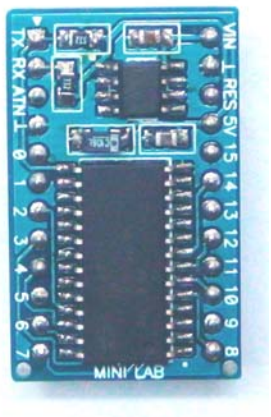


Table 2.1 shows the device pin assignments.

Table 2.1

Pin No.	Pin Name	Description
1	SDA/TX	I2C SDA signal (Serial output TX – for serial interface)
2	SCL/RX	I2C SCL signal (Serial input RX – for serial interface)

3	ATN	Attention: digital output (different functionality depending on the module configuration)
4	GND	Ground connection (same as pin 23)
5 - 20	0 - 15	General-purpose inputs/outputs pins (different functionality depending on the module configuration)
21	5V	5-volt DC input/output: if an unregulated voltage is applied to the VIN pin, then this pin will output 5 volts. If no voltage is applied to the VIN pin, then a regulated voltage 5V should be applied to this pin.
22	RES	Reset input: can be driven **HIGH** to force a reset. This pin is internally pulled **LOW** and may be left disconnected if not needed.
23	GND	Ground connection (same as pin 4)
24	VIN	Unregulated power in: accepts 7-15 VDC, witch is then internally regulated to 5 volts. Must be left unconnected if 5 volts is applied to the pin 21 (5V).

2.2 MiniLab communication interfaces.

The device supports 2 interfaces:

- I2C slave interface
- Serial TTL interface

I2C Slave Interface
Industry standard Philips I^2C bus compatible interface.
Data rate 100 kbps.

Serial Interface
Baud Rates 4800, 9600 and 19200 bits per second (default 9600 after power up or reset).
8 Bits per character
None Parity
1 Stop Bit
None Flow Control

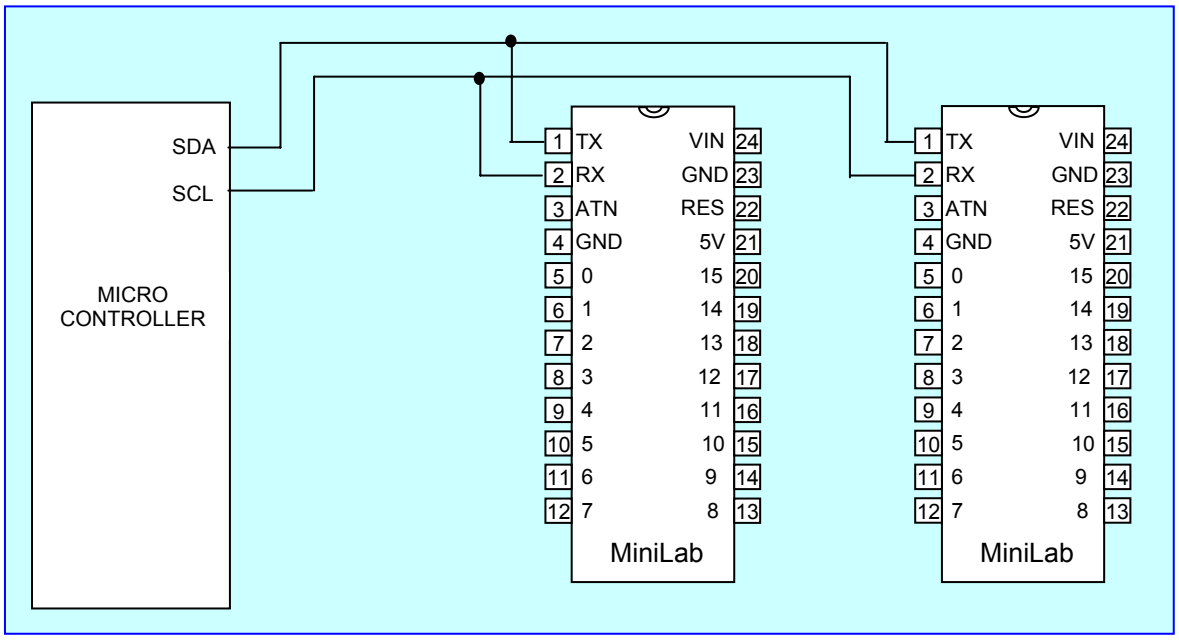
2.3 MiniLab technical parameters.

- Internal voltage regulator with input voltage 7 – 15 volts
- Voltage regulator MAX output current – 100mA
- 25 mA Sink on all digital Inputs/Outputs
- 30 mA on analog outputs
- Industrial Temperature Range -40C to +85C
- Board size: 1.3” x 0.8” (33mm x 20mm)
- 0.1” pin spacing
- 0.6” pin row distance

2.4 Connecting Multiple MiniLab devices.

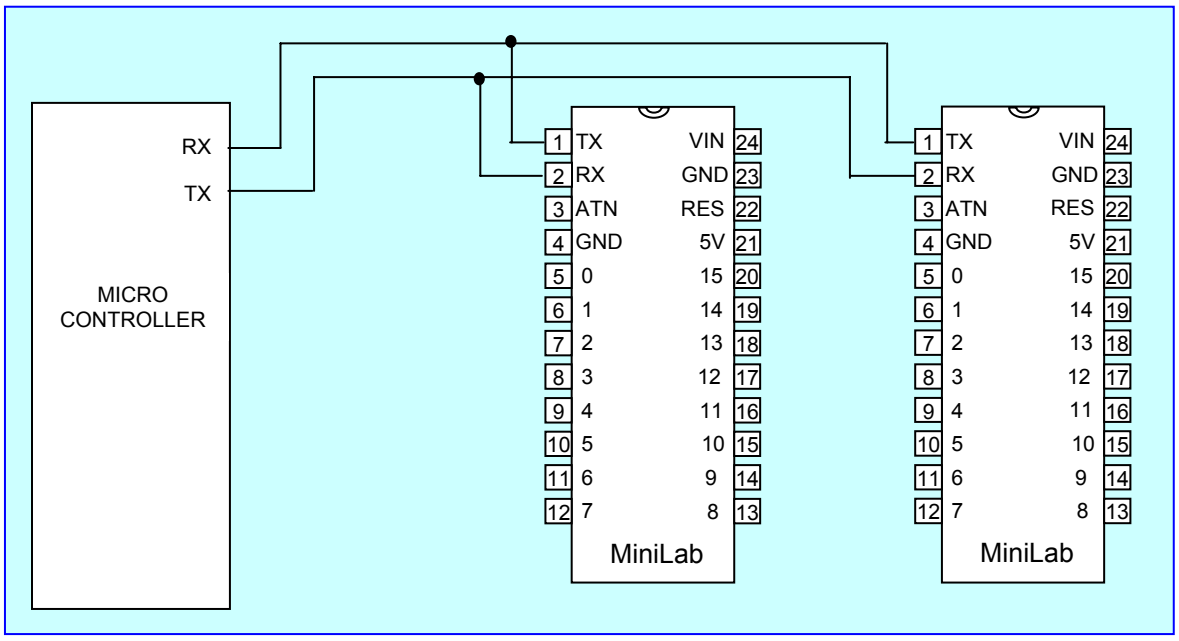
Each device must have its own unique address (ID). The address range is from ‘A’ to ‘Z’ (HEX from 0x41 to 0x5A). Default address shipped from the manufacture is ‘D’ (0x44).
The address can be easily changed by send the command “Set the new device address” (See the Title 3).

Connecting Multiple MiniLab devices with I2C Slave Interface.



NOTE: The MiniLab module includes the pull up resistors for SDA and SCL lines.

Connecting Multiple MiniLab devices with Serial TTL Interface.

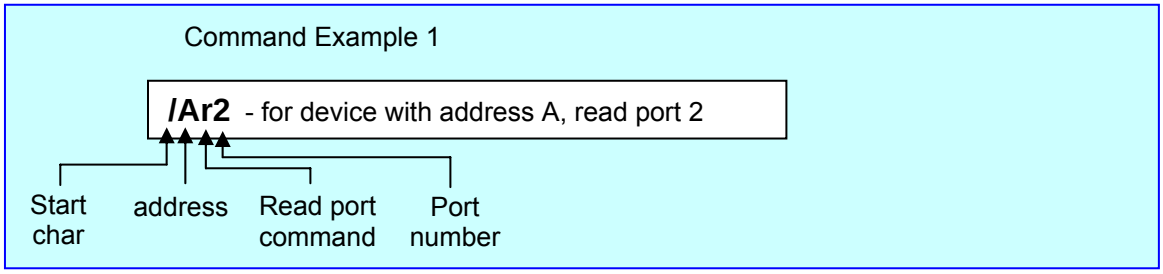


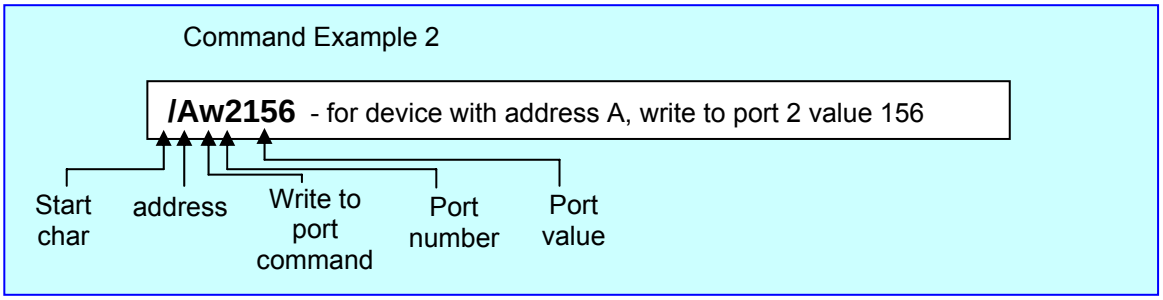
2.5 Command Format.

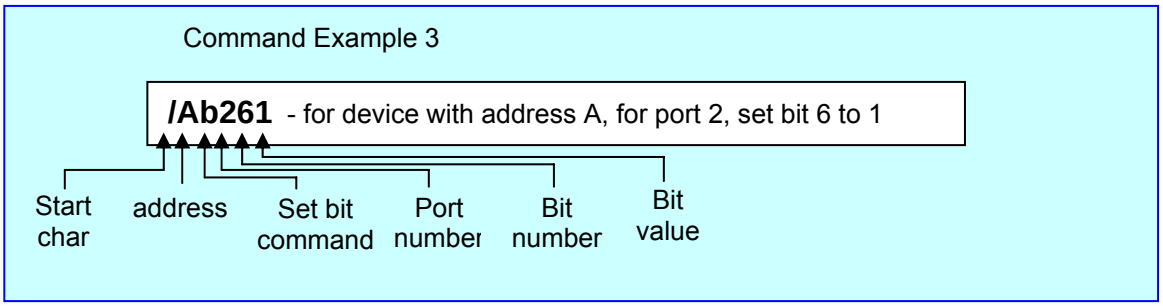
Commands to the MiniLab device are ASCII Character Strings.
All commands start with the '/' character and a single alpha device address. Then a one letter command code followed by the parameters. The command is terminated by the <CR> (carriage return) character.
Command Syntax:

1. **/<address><command><parameter1><CR>**
2. **/<address><command><parameter1><parameter2><CR>**
3. **/<address><command><parameter1><parameter2><parameter3><CR>**

Spaces or other punctuation characters are not allowed in the command character sequence.
ALL commands are CASE SENSITIVE.







3 PREPROGRAMMED CONFIGURATION DESCRIPTION.

MiniLab device includes 8 preprogrammed configurations. The configurations can be switched in run time. The dynamic reconfiguration time is less than 200 µs.
Every configuration has a command set. MiniLab includes also four common commands:

- Read the device information
- Set the new device address (ID)
- Set the configuration number
- Set the baud rate for serial port

No.	Command Description	Command Format	Parameters	Example
1	**I** – Read the device information	**/<addr>I<CR>**	/ - start char, <addr> - device address (ID), I – command char, <CR> - carriage return	**/AI** – read the information from device with address A. The device return message (8 chars): ML,A,2,1 Where ML – device type, A – device address, 2 – configuration number, 1 – revision number
2	**B** – Set the new device address	**/<addr>B<newaddr><CR>**	/ - start char, <addr> - device address, B – command char, <newaddr> - new address, <CR> - carriage return	**/ABD** – set for device with address A the new address D
3	**C** – Set the configuration number	**/<addr>C<newconf><CR>**	/ - start char, <addr> - device address, C – command char, <newconf> - configuration number, <CR> - carriage return	**/AC2** – set for device with address A the configuration number 2
4	**Y** – Set the baud rate for serial port	**/<addr>Y<rate><CR>**	/ - start char, <addr> - device address (if address = '0' 0x30 all devices accept this command), Y – command char, <rate> - baud rate for serial port (0 – 4800, 1 – 9600, 2 – 19200; default 9600 after power up or reset), <CR> - carriage return	**/0Y2** – set for all devices the baud rate 19200

Note: Wait 50ms after "Set new device address" command to write the data in internal flash memory, before send the next command.

3.1 Configuration 1. RC Servo Control.

In configuration 1 MiniLab device works as 16 channel servo controller. The device generates 16 continuous streams of pulses that are 500 to 2500 microseconds long, repeated fifty times per second. The pulse resolution is one microsecond.
Every servo has independent control. The device can set the position and speed for each servo, disable or enable servo in run time.

Fig.3.1 shows the device pin assignments for configuration 1.

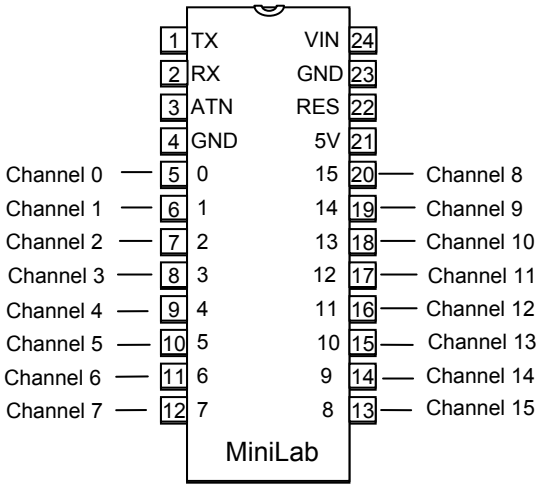


Fig.3.1

Command Set for Configuration 1.

No.	Command Description	Command Format	Parameters	Example
1	**A** – Set the servo position	**/<addr>A<ser><pos><CR>**	/ - start char, <addr> - device address (ID), A – command char, <ser> - servo number (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F), <pos> - position(500 to 2500), <CR> - carriage return	**/AA30850** – for device A set servo 3 position to 850.
2	**V** – Set the servo speed	**/<addr>V<ser><spd><CR>**	/ - start char, <addr> - device address, V – command char, <ser> - servo number (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F), <spd> - speed (0 to 255), <CR> - carriage return	**/AV2150** – for device A set servo 2 speed to 150
3	**J** – Disable servo	**/<addr>J<ser><CR>**	/ - start char, <addr> - device address, J – command char, <ser> - servo number (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F), <CR> - carriage return	**/AJ4** – for device with address A disable servo 4

Note: Set the servo position command enables the current RC servo.

3.2 Configuration 2. Data Acquisition System.

The configuration 2 includes the Data Acquisition System:

- 4 analog channels with programmable gain amplifier
- 14 bit Analog to Digital Converter (ADC)
- two 9 bit analog outputs (DAC)

Fig. 3.2 illustrates the system implementation insight in MiniLab device.

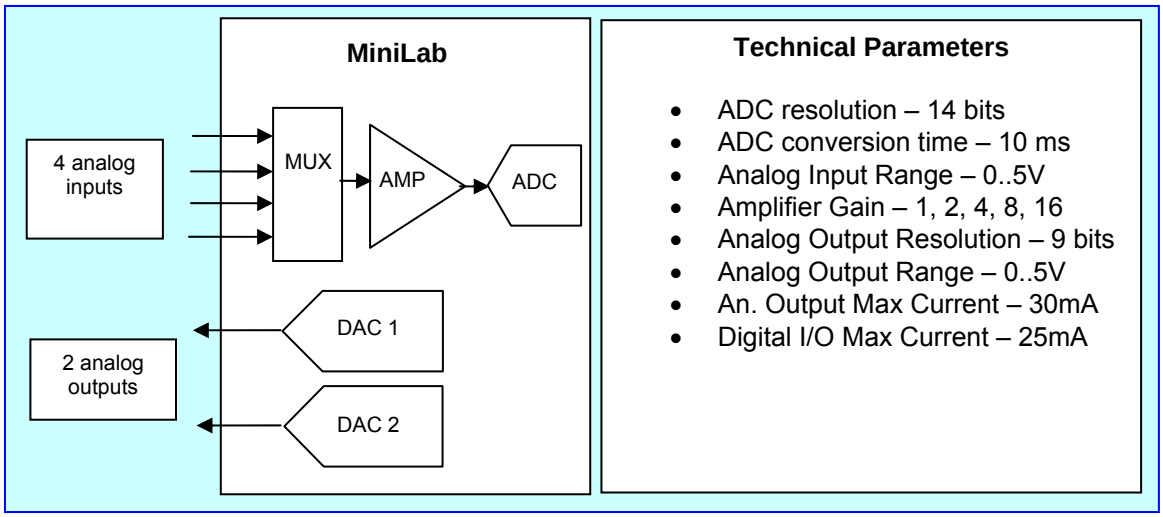
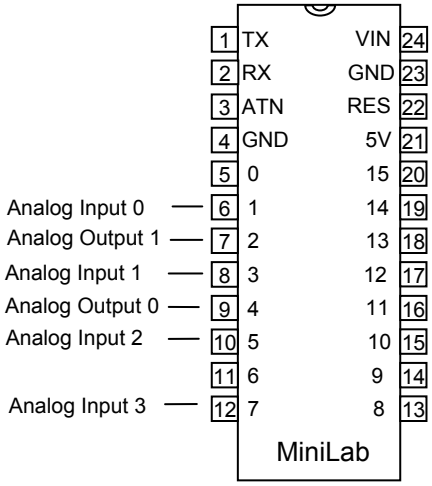


Fig.3.3 shows the device pin assignments for configuration 2.



Command Set for Configuration 2.

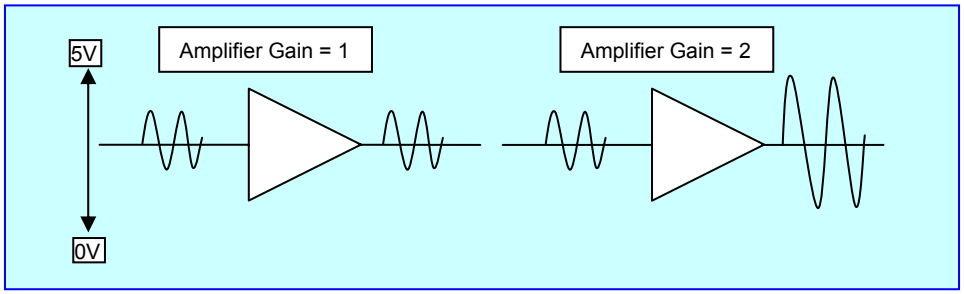
No.	Command Description	Command Format	Parameters	Example
1	**i** – Read analog input	**/<addr>i<ch><gain><CR>**	/ - start char, <addr> - device address (ID), i – command char, <ch> - channel (0 to 3), <gain> - amplifier gain (1 to 5), <CR> - carriage return	**/Ai13** – for device A read analog channel 1 with amplifier gain number 3. The device will return 5 chars of ADC value.
2	**t** – Write to analog output	**/<addr>V<ch><val><CR>**	/ - start char, <addr> - device address, t – command char, <ch> - channel (0,1), <val> - code value (000 to 511), <CR> - carriage return	**/At0215** – for device A write to analog output 0 value 215

Amplifier Gain Setup

Amplifier Gain Number	Amplifier Gain Value
1	1
2	2
3	4
4	8
5	16

Note: Analog signal measurement.

The MiniLab device operates on a single power VDD =5 volts. Analog signals in most systems are typically of both positive and negative polarity around some reference or ground. The MiniLab only handles signals of positive polarity with respect to VDD. An artificial ground is constructed on the chip to provide a reference point for signals of both polarities; this reference is called Analog Ground = VDD/2.



The expected ADC code is 0 for 0 volts, 8192 for Analog Ground and 16383 for 5 volts.
The programmed gain amplifier has not rail to rail input. The calibration procedure is recommended for precision measurement:

1. Connect the reference voltage Vref1 to the analog input and measure the ADC code ADC1.
2. Connect the reference voltage Vref2 to the analog input and measure the ADC code ADC2.
3. Use the formula to calculate input voltage Vx :

   Vx = Vref1 + (ADCx – ADC1)*K, where K = (Vref2 – Vref1) / (ADC2 – ADC1).

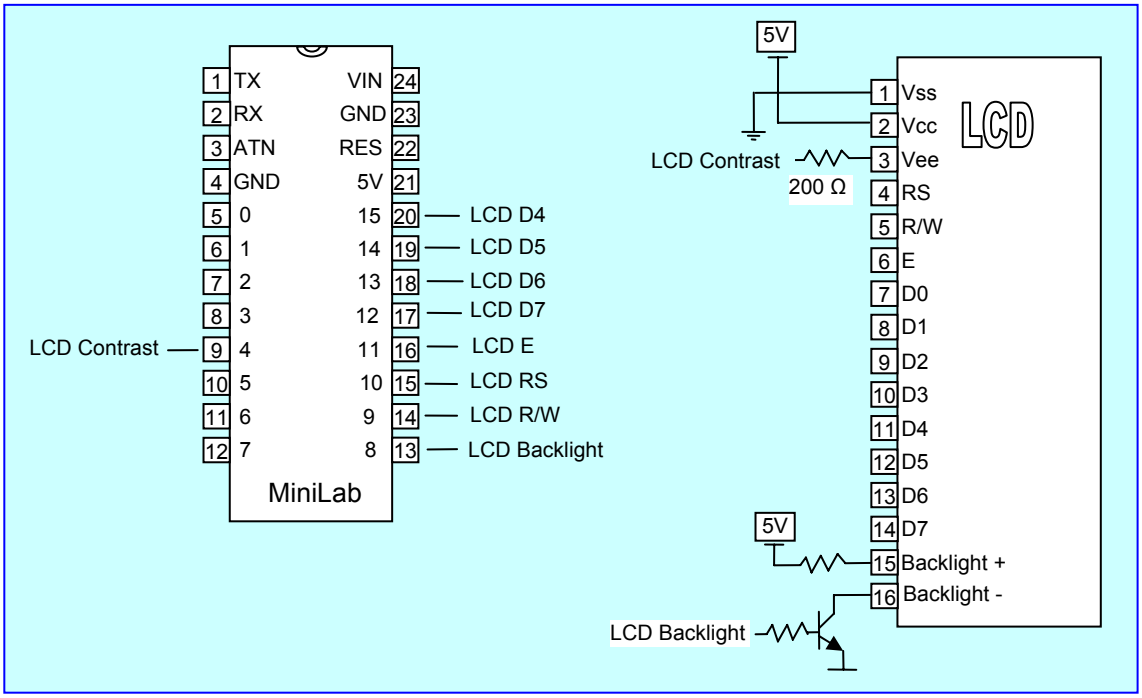
   Example:
   Vref1 = 1V; ADC1 = 3200;
   Vref2 = 4V; ADC2 = 12700;

   For ADCx = 5340 Vx = 1 + (5340-3200) * (4 – 1) / (12700 – 3200) = 1.6758 V.

3.3 Configuration 3. Character LCD control.

The configuration 3 supports the character LCD from LCD 8x1 to LCD 20x4.

Fig.3.4 shows the device pin assignments for configuration 3 and LCD connections.



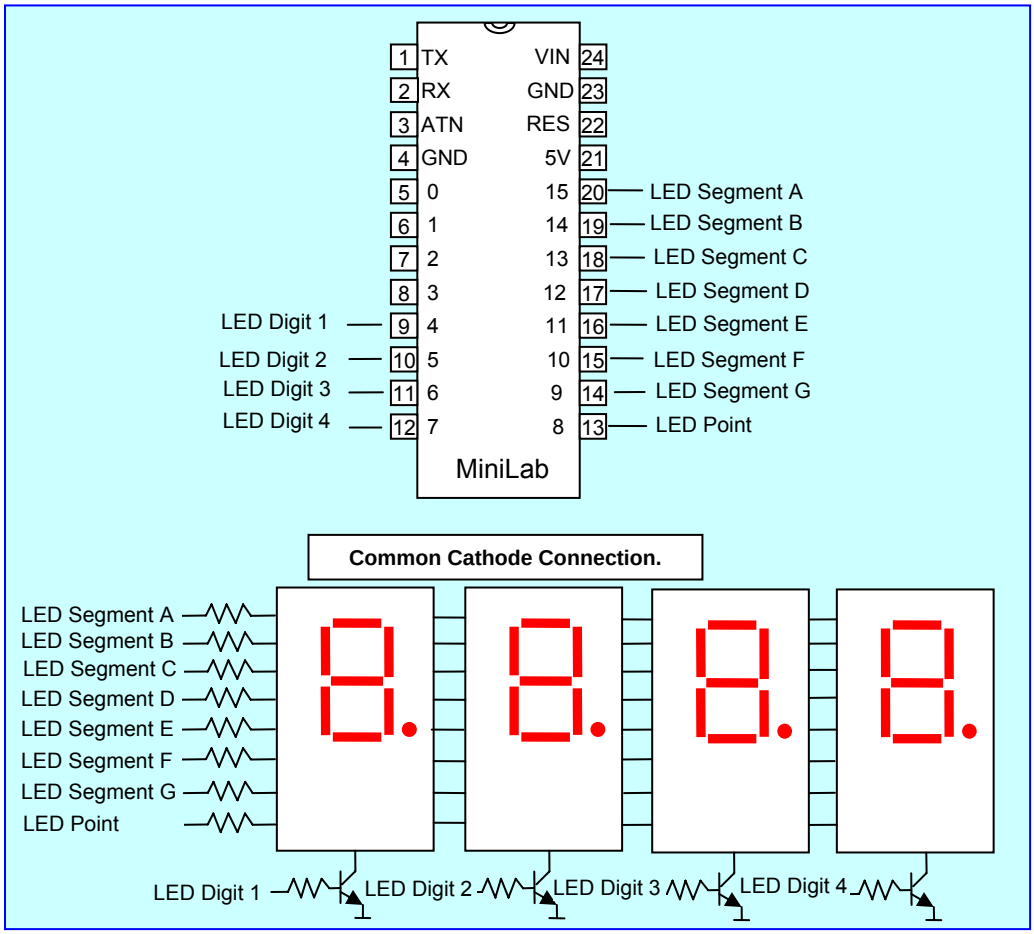
Command Set for Configurarion 3.

No.	Command Description	Command Format	Parameters	Example
1	**c** – Clear LCD	**/<addr>c<CR>**	/ - start char, <addr> - device address (ID), c – command char, <CR> - carriage return	**/Ac** – for device A clear LCD.
2	**p** – Set the cursor position LCD	**/<addr>p<str><pos><CR>**	/ - start char, <addr> - device address, p – command char, <str> - string number(0,1,2,3), <pos> - cursor position (00 to 19), <CR> - carriage return	**/Ap215** – for device A set LCD cursor: - string 2 - position 15
3	**t** – LCD print string	**/<addr>t<str><CR>**	/ - start char, <addr> - device address, t – command char, <str> - string (up to 20 chars), <CR> - carriage return	**/AtHello** – for device with address A print string “Hello” at current cursor position
4	**x** – Set LCD contrast	**/<addr>x<val><CR>**	/ - start char, <addr> - device address, x – command char, <val> - contrast value (1…255), <CR> - carriage return	**/Ax35** – for device with address A set LCD contrast value 35
5	**b** – LCD set the backlight	**/<addr>b<val><CR>**	/ - start char, <addr> - device address, b – command char, <val> - backlight Value (0 – 250), <CR> - carriage return	**/Ab50** – for device with address A set LCD backlight value 50

3.4 Configuration 4. LED 7-Segment Display control.

The configuration 4 supports Single-Digit, Dual-Digit, Triple-Digit, Quad-Digit Displays.

Fig.3.5 shows the device pin assignments for configuration 4 and LED connections.



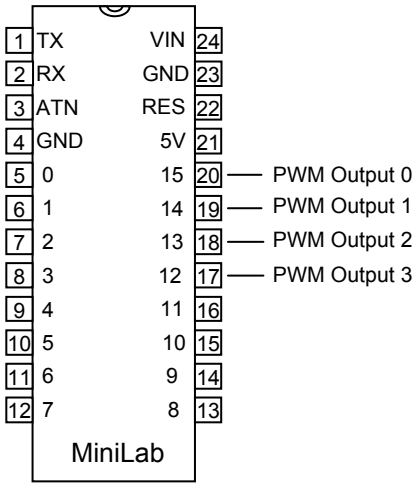
Command Set for Configuration 4.

No.	Command Description	Command Format	Parameters	Example
1	**p** – Set LED poit position	**/<addr>p<s><d><CR>**	/ - start char, <addr> - device address (ID), p – command char, <s> - enable poit display (0 – off, 1 – on), <d> - digit location (1…4), <CR> - carriage return	**/Ap12** – for device A set LED point for digit 2.
2	**t** – LED print string	**/<addr>t<p><l><v><CR>**	/ - start char, <addr> - device address, t – command char, <p> - start digit position (1…4), <l> - length of digit to display (1…4), <v> - value to display (0…9999), <CR> - carriage return	**/At141234** – for device with address A print string “12.34”.

3.5 Configuration 5. DC Motor Control.

The configuration 5 provides four 8 bit PWM blocks to control up to 4 DC motor.

Fig.3.6 shows the device pin assignments for configuration 5.



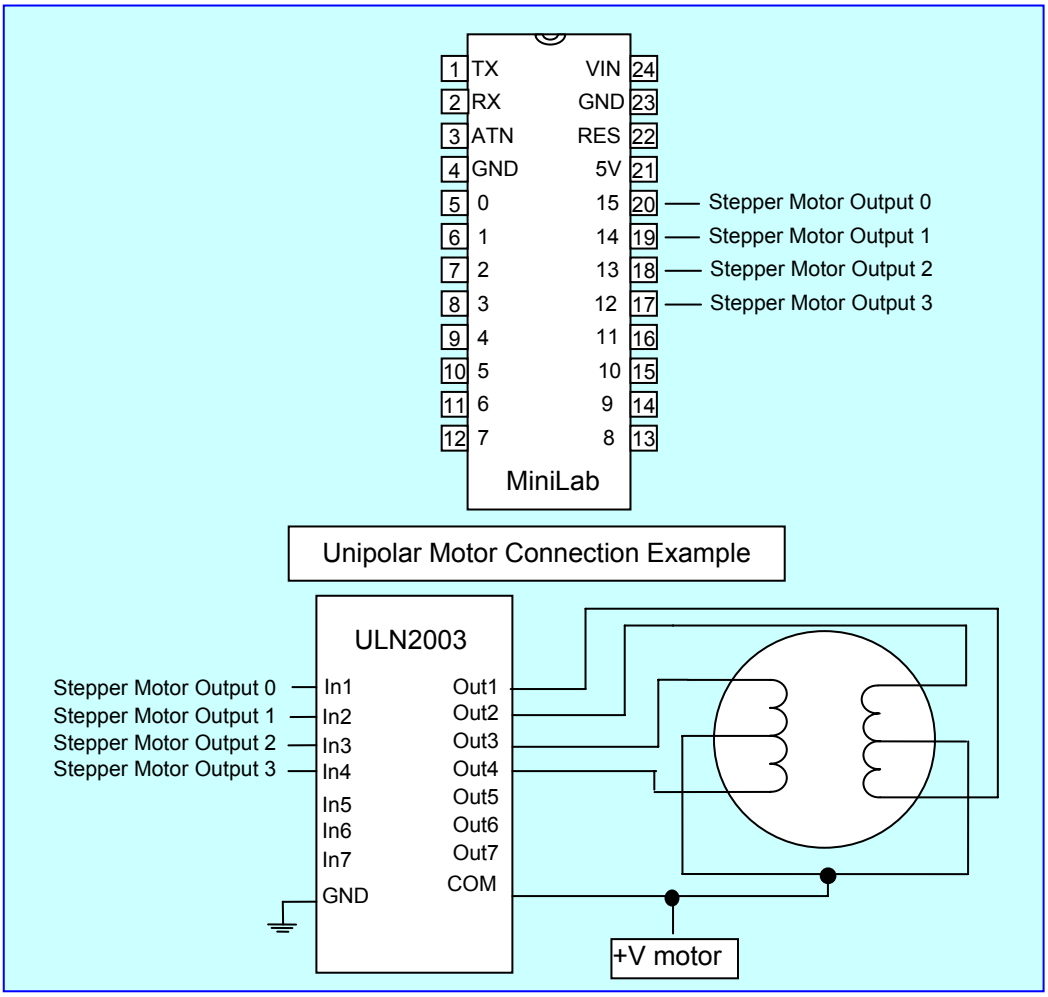
Command for Configuration 5.

No.	Command Description	Command Format	Parameters	Example
1	**p** – Set PWM pulse width	**/<addr>p<ch><val><CR>**	/ - start char, <addr> - device address (ID), p – command char, <ch> - PWM number (0 to 3), <val> - pulse width (0 to 255, 0 – disable PWM), <CR> - carriage return	**/Ap2150** – for device A set PWM 2 pulse width to 150.

3.6 Configuration 6. Stepper Motor Control.

The configuration 6 provides the stepper motor control (current version support only unipolar motors).

Fig.3.7 shows the device pin assignments for configuration 6.

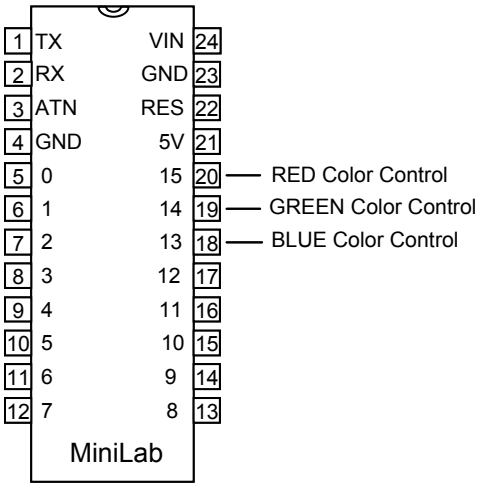


Command Set for Configuration 6.

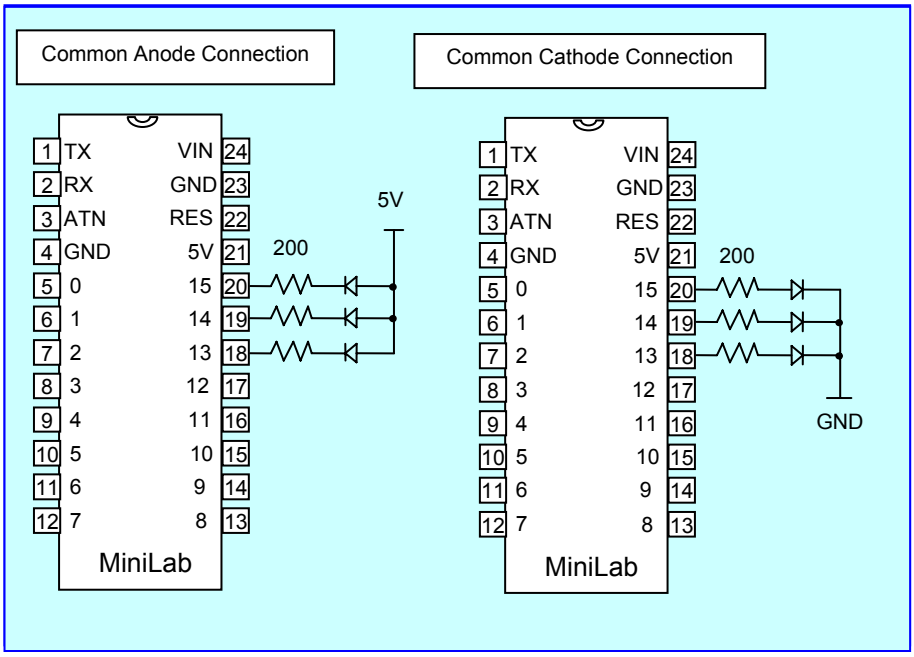
No.	Command Description	Command Format	Parameters	Example
1	**j** – Stepper Motor Setup	**/<addr>j<mt><mode><CR>**	/ - start char, <addr> - device address (ID), j – command char, <mt> - motor type (0 – unipolar, 1 - bipolar), <mode> - motor mode (0 – full step, 1 - half step), <CR> - carriage return	**/Aj01** – for device A setup the unipolar motor in half step mode.
2	**n** – Set the number of steps	**/<addr>n<stps><CR>**	/ - start char, <addr> - device address, n – command char, <stps> - number of steps (-32767 to 32767), <CR> - carriage return	**/An1500** – for device with address A make 1500 steps.
3	**t** – Set the step delay	**/<addr>t<dly><CR>**	/ - start char, <addr> - device address, t – command char, <dly> - step delay (1 to 1000 ms), <CR> - carriage return	**/At132** – for device with address A set the step delay 132 ms.

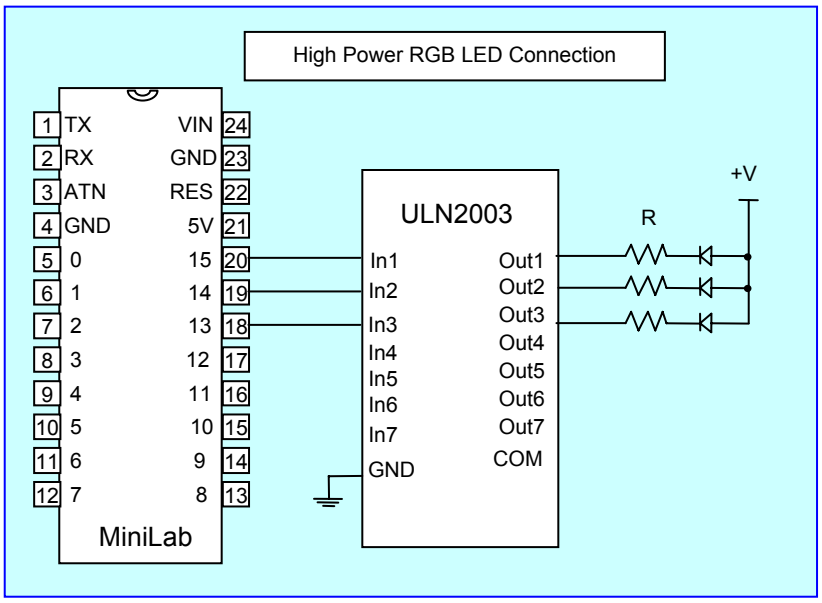
3.7 Configuration 7. RGB LED Control.

The configuration 7 provides the 24 bit color control for RGB LED.
Fig.3.8 shows the device pin assignments for configuration 7.



RGB LED Connection





Command Set for Configuration 7.

No.	Command Description	Command Format	Parameters	Example
1	**t** – Set LED type	**/<addr>t<type><CR>**	/ - start char, <addr> - device address (ID), t – command char, <type> - LED type (0 – common anode, 1 – common cathode), <CR> - carriage return	**/At1** – for device A set the LED type – common cathode.
2	**r** – Set the RED color value	**/<addr>r<value><CR>**	/ - start char, <addr> - device address, r – command char, <value> - RED color value (0 to 255), <CR> - carriage return	**/Ar150** – for device with address A set the RED color value 150.
3	**g** – Set the GREEN color value	**/<addr>g<value><CR>**	/ - start char, <addr> - device address, g – command char, <value> - GREEN color value (0 to 255), <CR> - carriage return	**/Ag132** – for device with address A set the GREEN color value 132.
4	**b** – Set the GREEN color value	**/<addr>b<value><CR>**	/ - start char, <addr> - device address, b – command char, <value> - BLUE color value (0 to 255), <CR> - carriage return	**/Ab45** – for device with address A set the BLUE color value 45.

3.8 Configuration 8. Analog Tool Set.

The configuration 8 includes the analog tool set:

- 14 bit Analog to Digital Converter (ADC)
- 9 bit analog output (DAC)
- Two the programmable gain amplifiers (Amplifier 1 output connected to ADC input)
- Comparator with programmable reference level

Fig. 3.9 illustrates the system implementation insight in MiniLab device.

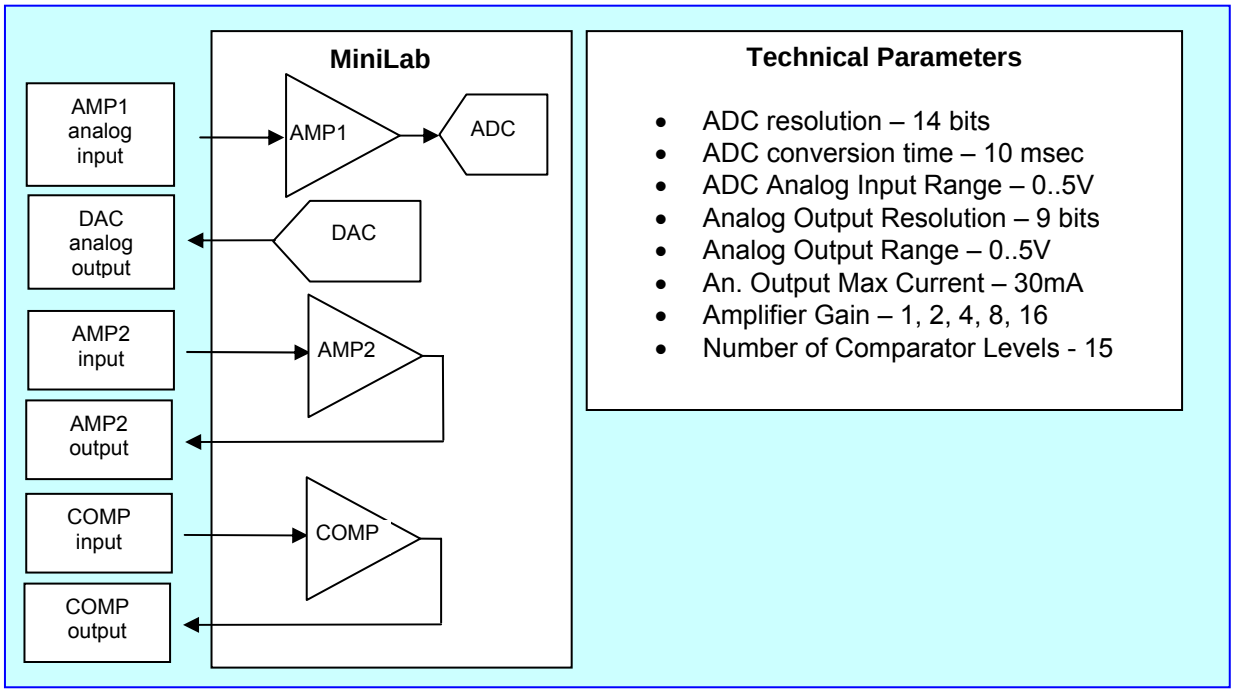
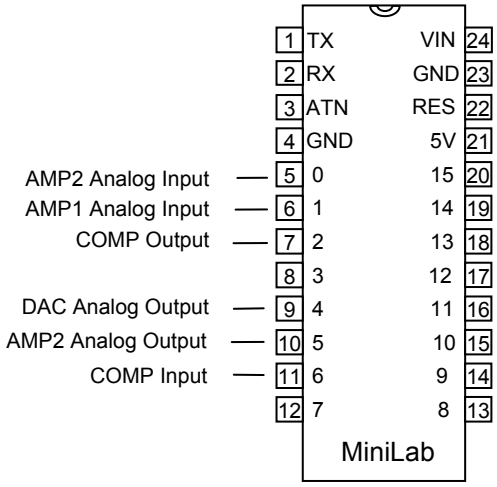


Fig.3.10 shows the device pin assignments for configuration 8.



Command Set for Configuration 8.

No.	Command Description	Command Format	Parameters	Example
1	**i** – Read analog input	**/<addr>i<gain><CR>**	/ - start char, <addr> - device address (ID), i – command char, <gain> - amplifier gain (1 to 5), <CR> - carriage return	**/Ai3** – for device A read ADC value with amplifier gain number 3. The device will return 5 chars of ADC value.
2	**t** – Write to analog output	**/<addr>t<val><CR>**	/ - start char, <addr> - device address, t – command char, <val> - code value (000 to 511), <CR> - carriage return	**/At15** – for device A write to analog output value 215
3	**g** – Set Amplifier 2 Gain	**/<addr>g<gain><CR>**	/ - start char, <addr> - device address (ID), g – command char, <gain> - amplifier gain (1 to 5), <CR> - carriage return	**/Ag3** – for device A set the amplifier gain number 3.
4	**f** – Set Comparator Level	**/<addr>f<level><CR>**	/ - start char, <addr> - device address (ID), f – command char, <level> -comparator level (1 to 16), <CR> - carriage return	**/Af5** – for device A set the comparator level number 5.

Amplifier Gain Setup

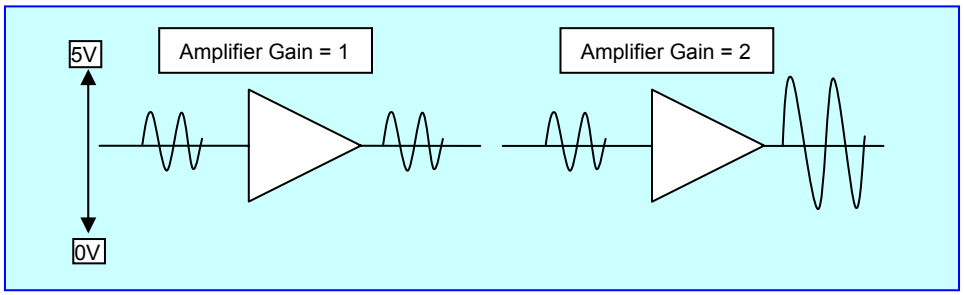
Amplifier Gain Number	Amplifier Gain Value
1	1
2	2
3	4
4	8
5	16

Comparator Level Setup

Comparator Level Number	Comparator Level Value
1	0.31 Volt
2	0.625 Volt
3	0.94 Volt
4	1.25 Volt
5	1.56 Volt
6	1.875 Volt
7	2.185 Volt
8	2.50 Volt
9	2.81 Volt
10	3.125 Volt
11	3.44 Volt
12	3.75 Volt
13	4.06 Volt
14	4.375 Volt
15	4.685 Volt

Note: Analog signal measurement.

The MiniLab device operates on a single power VDD =5 volts. Analog signals in most systems are typically of both positive and negative polarity around some reference or ground. The MiniLab only handles signals of positive polarity with respect to VDD. An artificial ground is constructed on the chip to provide a reference point for signals of both polarities; this reference is called Analog Ground = VDD/2.



The expected ADC code is 0 for 0 volts, 1023 for Analog Ground and 2047 for 5 volts.
The programmed gain amplifier has not rail to rail input. The calibration procedure is recommended for precision measurement:

4. Connect the reference voltage Vref1 to the analog input and measure the ADC code ADC1.
5. Connect the reference voltage Vref2 to the analog input and measure the ADC code ADC2.
6. Use the formula to calculate input voltage Vx :

   Vx = Vref1 + (ADCx – ADC1)*K, where K = (Vref2 – Vref1) / (ADC2 – ADC1).

4 Appendix A: I2C communication with BASIC Stamp.

BASIC Stamp Modules (BS2p, BS2pe, BS2px) by Parallax, Inc. use two commands to communicate with I2C devices: I2CIN and I2COUT (See *BASIC Stamp Syntax and Reference Manual* for command descriptions).
The commands use the SlaveID variable indicating the unique ID of the I2C chip. The SlaveID is 8-bit pattern whose upper 7-bits contain the unique ID of the device you wish to communicate with. The lowest bit indicates whether this is a write operation (0) or a read operation (1).

7	6	5	4	3	2	1	0
A6	A5	A4	A3	A2	A1	A0	R/W

To convert I2C device ID to SlaveID need shift to left I2C device ID and add 1 for read operation. The next table shows the I2C device ID and SlaveID value.

FLEXEL I2C ID	‘A’ (0x41)	‘B’ (0x42)	‘C’ (0x43)	‘D’ (0x44)	‘E’ (0x45)	‘F’ (0x46)	‘G’ (0x47)	‘0’ (0x30)
SlaveID Write	0x82	0x84	0x86	0x88	0x8A	0x8C	0x8E	0x60
SlaveID Read	0x83	0x85	0x87	0x89	0x8B	0x8D	0x8F	0x61

```
‘An example program Read analog input (send the MiniLab command ‘/Ai13’)

#IF ($STAMP < BS2P) #THEN
   #ERROR “Program requires BS2p, BS2pe, or BS2px.”
#ENDIF
‘*****************************************************************************
SDA PIN 8                       ‘I2C SDA pin
SCL PIN 9                       ‘I2C SCL pin
slvAddrWR   CON   $82           ‘ I2C write address (MiniLab address ‘A’)
slvAddrRD   CON   $83           ‘ I2C read address (MiniLab address ‘A’)
Darr          VAR   Byte(5)     ‘ data array

MAIN:
   PAUSE 100

   Darr(0) = $2F      ‘ ‘/’ – char
   Darr(1) = $41      ‘ ‘A’ – char
   Darr(2) = $69      ‘ ‘i’ – char
   Darr(3) = $31      ‘ ‘1’ – char
   Darr(4) = $33      ‘ ‘3’ – char
   Darr(5) = CR       ‘ CR’ – carriage return

   I2COUT SDA, slvAddrWR, [Darr(0), Darr(1), Darr(2), Darr(3), Darr(4),Darr(5)]  ‘ send ‘/Ai13’
   PAUSE 15     ‘ pause 15 ms for ADC data conversion
   I2CIN SDA, slvAddrRD, [STR Darr\5]    ‘ read 5 chars

END
```