

**Preprogrammed dynamically reconfigurable device
«FLEXEL-1»**

User manual

CONTENTS

1	INTRODUCTION.....	3
2	FLEXEL DESCRIPTION.	4
2.1	FLEXEL package and pin assignments.	4
2.2	FLEXEL communication interfaces.	5
2.3	FLEXEL technical parameters.	5
2.4	Connecting Multiple FLEXEL devices.	6
2.5	Command Format.	7
3	PREPROGRAMMED CONFIGURATION DESCRIPTION.....	8
3.1	Configuration 1. RC Servo Control.....	9
3.2	Configuration 2. Data Acquisition System.....	10
3.3	Configuration 3. Character LCD and Keypad control.	13
3.4	Configuration 4. LED 7-Segment Display and Keypad control.....	14
3.5	Configuration 5. DC Motor Control.....	16
3.6	Configuration 6. Stepper Motor Control.	17
4	APPENDIX A: I2C COMMUNICATION WITH BASIC STAMP....	19

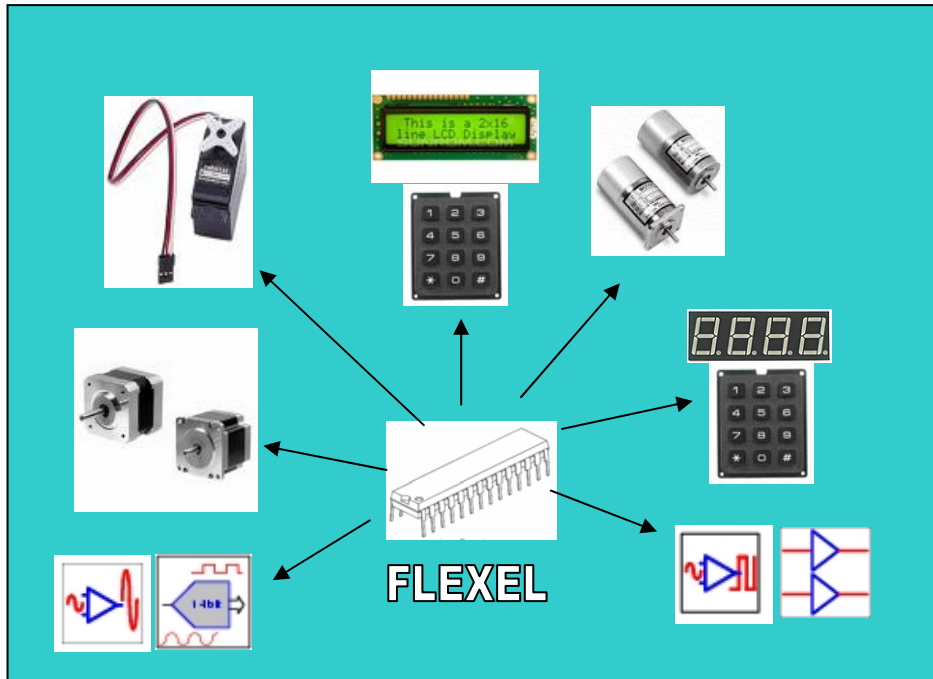
Copyrights

This documentation is copyright 2008 by IMS, LLC. By downloading or obtaining a printed copy of this documentation or software you agree that it is to be used exclusively with IMS products. Any other uses are not permitted and may represent a violation of Integrated Micro Systems copyrights, legally punishable according to Federal copyright or intellectual property laws.

Disclaimer of Liability

IMS, LLC is not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, or any costs of recovering, reprogramming, or reproducing any data stored in or used with IMS products. IMS is also not responsible for any personal damage, including that to life and health, resulting from use of any of our products.

1 Introduction.



FLEXEL 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.

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

The device supports 3 interfaces:

- I2C slave interface
- Serial TTL interface
- Serial TTL inverted interface (connection to BASIC STAMP)

The device includes 6 preprogrammed configurations. The configurations can be switched in run time. The dynamic reconfiguration time is less than 200 μ s.

FLEXEL 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, 8 digital inputs/outputs

3	Character LCD and 4x4 matrix Keypad control
4	LED 7-Segment Display and Keypad control (supports Single-Digit, Dual-Digit, Triple-Digit, Quad-Digit Displays and 4x4 Keypad)
5	Up to 4 DC motor control (includes four 8 bit PWM blocks), 16 digital inputs/outputs
6	Stepper motor control, 16 digital inputs/outputs

2 FLEXEL DESCRIPTION.

2.1 FLEXEL package and pin assignments.

The FLEXEL devices are available in 28-pin PDIP (300 mil) packages.

Most the I/O Pins have different functionality depending on the module configuration.

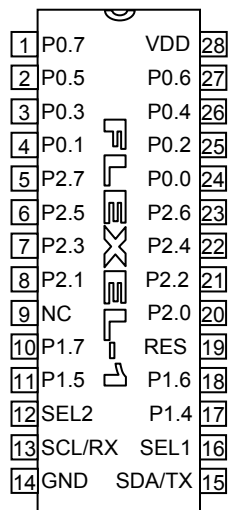


Table 2.1 shows the device pin assignments.

Table 2.1

Pin No.	Pin Name	Description
1	P0.7	Port 0, Bit 7
2	P0.5	Port 0, Bit 5
3	P0.3	Port 0, Bit 3
4	P0.1	Port 0, Bit 1
5	P2.7	Port 2, Bit 7
6	P2.5	Port 2, Bit 5
7	P2.3	Port 2, Bit 3
8	P2.1	Port 2, Bit 1
9	NC	Not connected
10	P1.7	Port 1, Bit 7
11	P1.5	Port 1, Bit 5
12	SEL2	Select the communication interface , Bit 2
13	SCL/RX	I2C SCL signal (RX – for serial interface)
14	GND	Ground connection
15	SDA/TX	I2C SDA signal (TX – for serial interface)
16	SEL1	Select the communication protocol , Bit 1
17	P1.4	Port 1, Bit 4

18	P1.6	Port 1, Bit 6
19	RES	Active high pin reset with internal pull down.
20	P2.0	Port 2, Bit 0
21	P2.2	Port 2, Bit 2
22	P2.4	Port 2, Bit 4
23	P2.6	Port 2, Bit 6
24	P0.0	Port 0, Bit 0
25	P0.2	Port 0, Bit 2
26	P0.4	Port 0, Bit 4
27	P0.6	Port 0, Bit 6
28	VDD	Supply voltage

2.2 FLEXEL communication interfaces.

The device supports 3 interfaces:

- I2C slave interface
- Serial TTL interface
- Serial TTL inverted interface (connection to BASIC STAMP).

I2C Slave Interface

Industry standard Philips I²C 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

The communication interface is determined by the states of the SEL1 (pin 16) and SEL2 (pin 12) signals. These pins have internal pull up resistors. Table 2.2 shows the communication protocol selection.

Table 2.2.

Interface	SEL1	SEL2
I2C Slave	Not connected	Not connected
Serial TTL	Connected to Ground	Not connected
Serial TTL inverted	Not connected	Connected to Ground

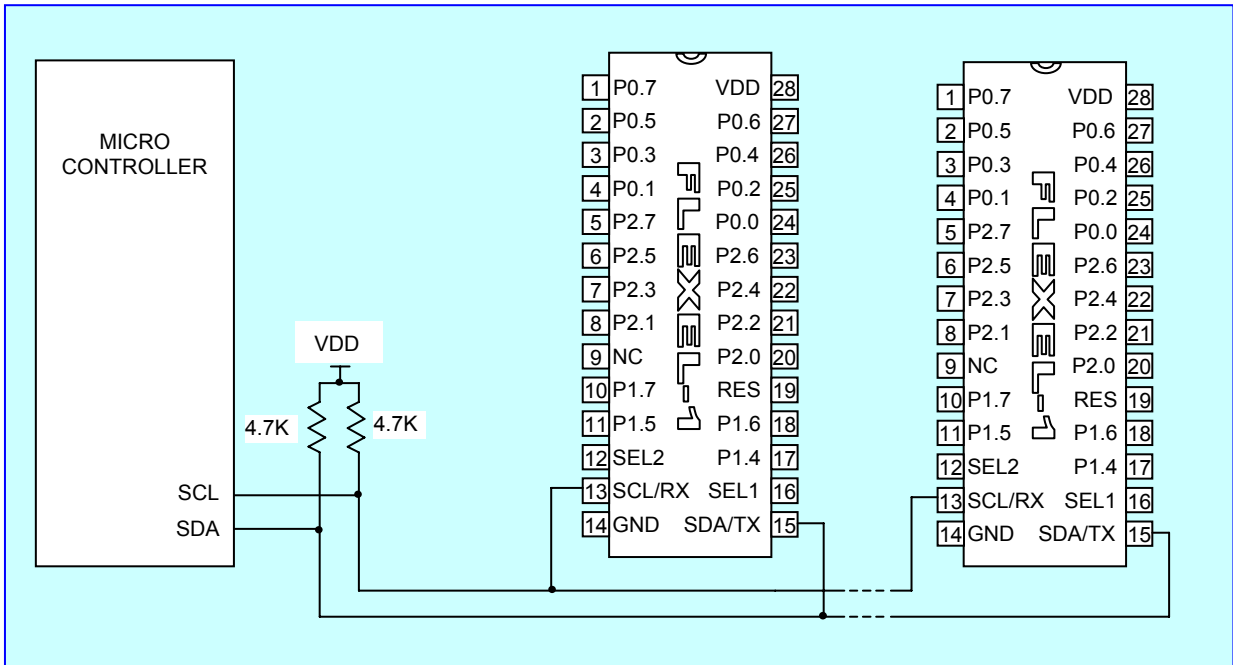
2.3 FLEXEL technical parameters.

- 3.3V (FLEXEL-1A) and 5V Operating Voltage
- 24 MHz Processor Speed
- 24 MHz internal Oscillator
- 25 mA Sink on all digital Inputs/Outputs
- 30 mA on analog outputs
- Industrial Temperature Range -40C to +85C

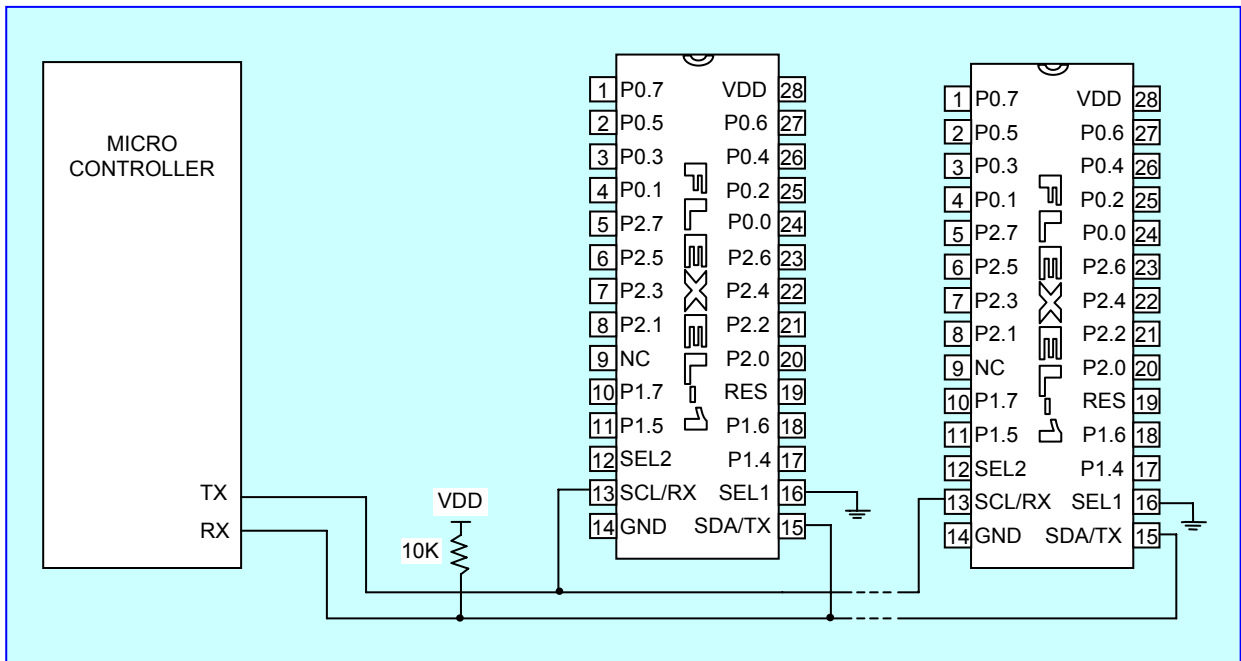
2.4 Connecting Multiple FLEXEL devices.

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 'A' (0x41). The address can be easily changed by send the command "Set the new device address" (See the Title 3).

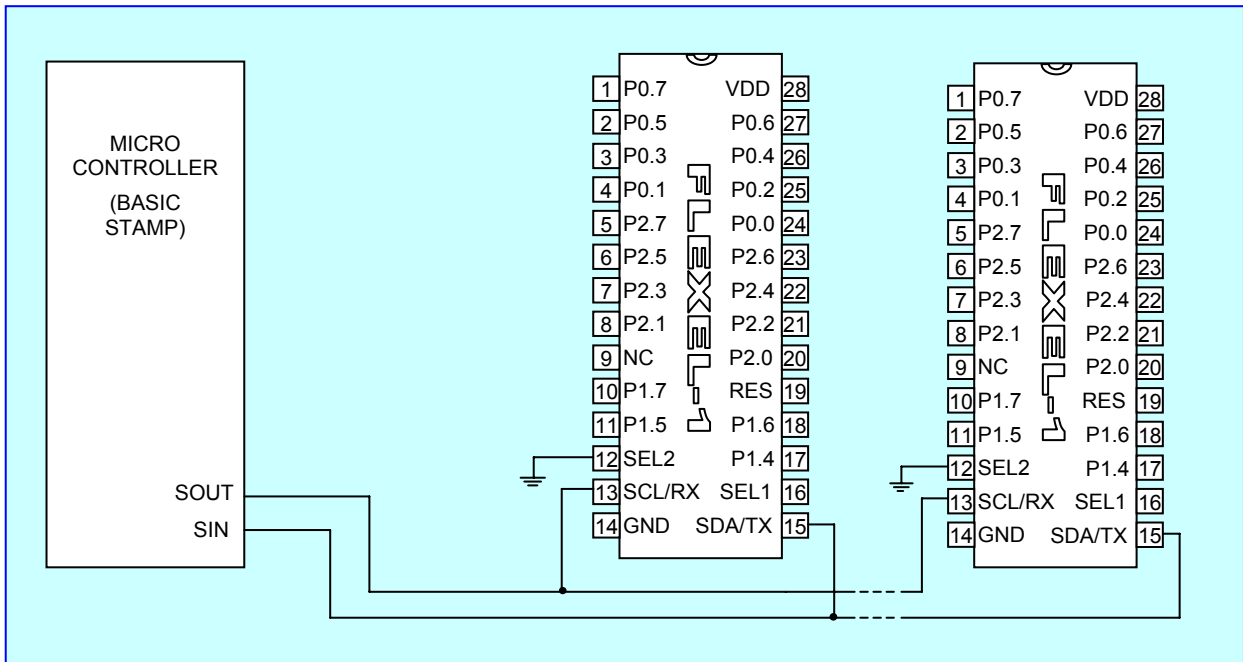
Connecting Multiple FLEXEL devices with I2C Slave Interface.



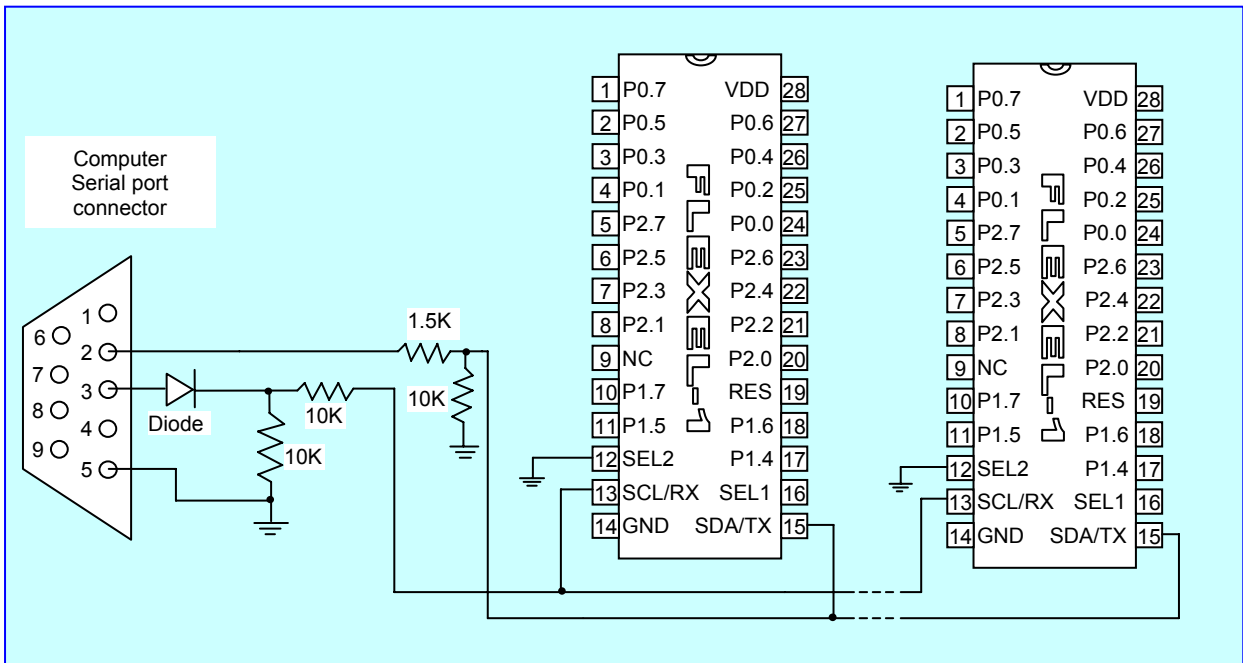
Connecting Multiple FLEXEL devices with Serial TTL Interface.



Connecting Multiple FLEXEL devices with Serial TTL Inverted Interface.



Connecting Multiple FLEXEL devices to Computer Serial port.



2.5 Command Format.

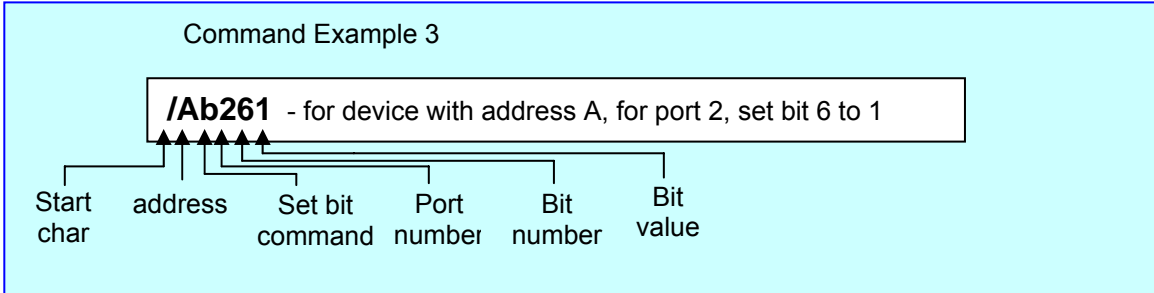
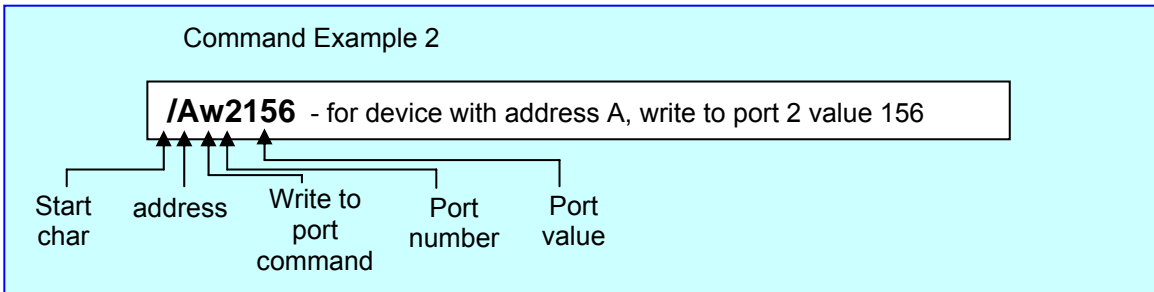
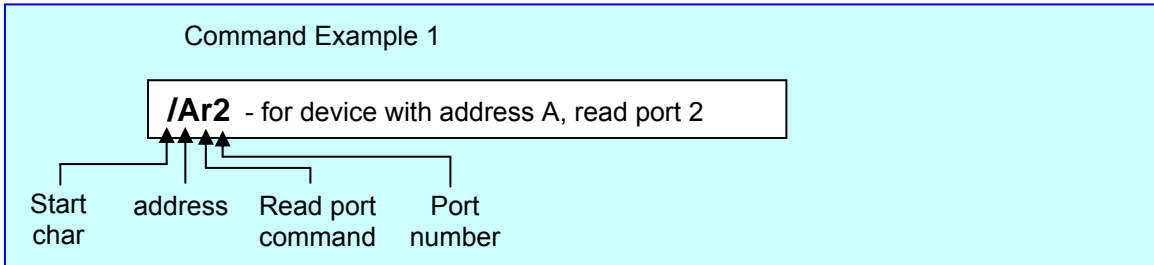
Commands to the FLEXEL device are ASCII Character Strings.

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.

FLEXEL device includes 6 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. FLEXEL 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 (9 chars): F01,A,2,1

				Where F01 – 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
3	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	/0YA2 – set for all devices the baud rate 19200

Note: Wait 50ms after “Set new device address” and “Set the configuration number” commands to write the data in internal flash memory, before send the next command.

3.1 Configuration 1. RC Servo Control.

In configuration 1 FLEXEL 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. It provides very accurate control of the servo position.

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 in configuration 1.

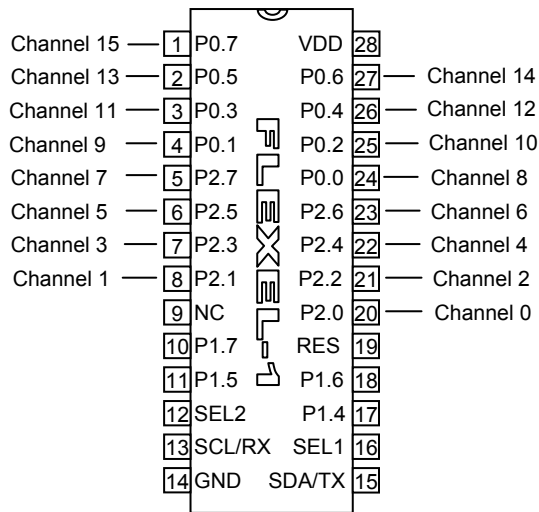


Fig.3.1

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

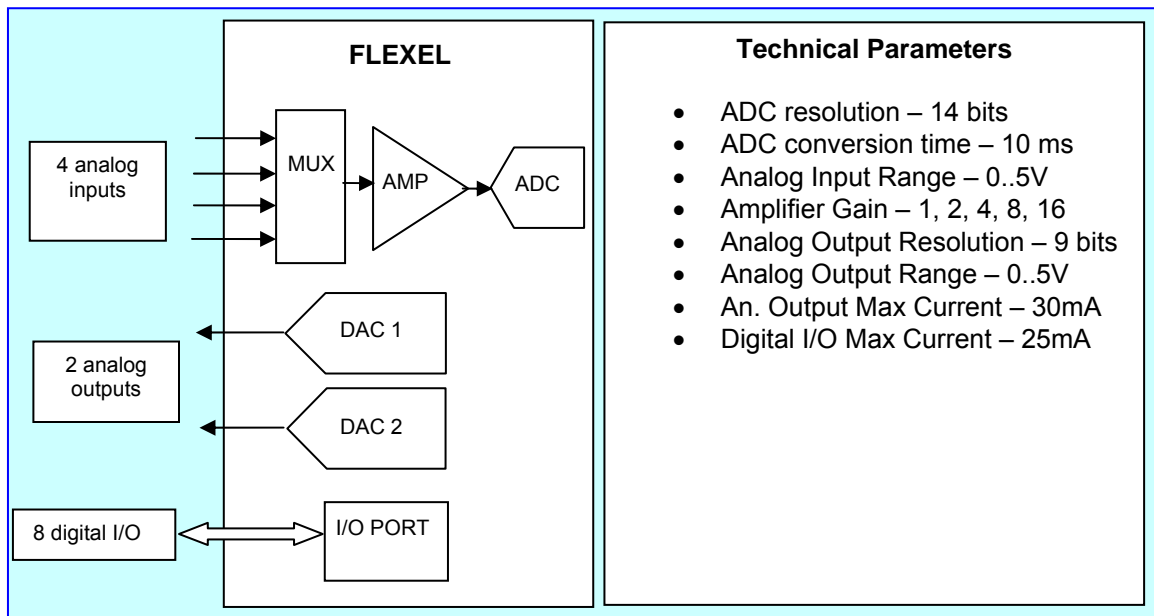
Note: Set the servo position command enables the current RC servo.

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)
- 8 digital inputs/outputs

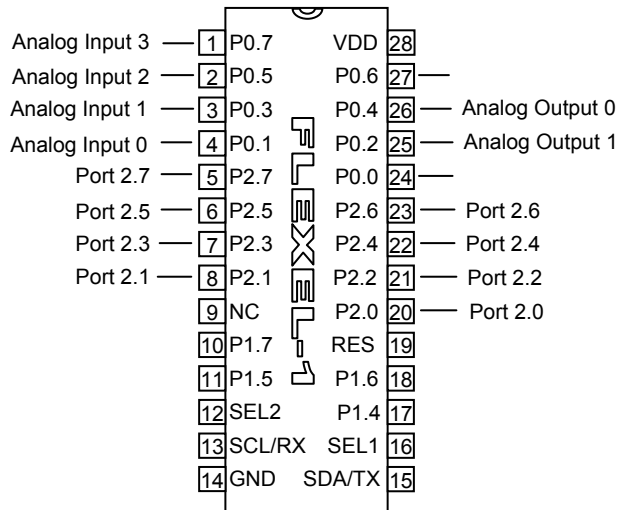
Fig. 3.2 illustrates the system implementation insight in FLEXEL device.



Technical Parameters

- ADC resolution – 14 bits
- ADC conversion time – 10 ms
- Analog Input Range – 0..5V
- Amplifier Gain – 1, 2, 4, 8, 16
- Analog Output Resolution – 9 bits
- Analog Output Range – 0..5V
- An. Output Max Current – 30mA
- Digital I/O Max Current – 25mA

Fig.3.3 shows the device pin assignments in 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
3	d – Set port direction	 /<addr>d<prt><dir><CR>	/ - start char, <addr> - device address, d – command char, <prt> - port number (2), <dir> - direction (0 – input, 1 – output), <CR> - carriage return	/Ad21 – for device with address A set port 2 direction to output
4	r – Read port	 /<addr>r<prt><CR>	/ - start char, <addr> - device address, r – command char, <prt> - port number (2), <CR> - carriage return	/Ar2 – for device with address A read port 2. The device will return 3 chars of port value.
5	w – Write to port	 /<addr>w<prt><val><CR>	/ - start char, <addr> - device address, w – command char, <prt> - port number (2), <val> - port value, <CR> - carriage return	/Aw2123 – for device with address A write to port 2 value 123
6	u – Set bit direction	 /<addr>u<p><d><CR>	/ - start char, <addr> - device address, u – command char, <p> - port number (2), - bit number (0 to 7), <d> - direction (0 – input, 1 –	/Au261 – for device with address A set port 2 bit 6 direction to output

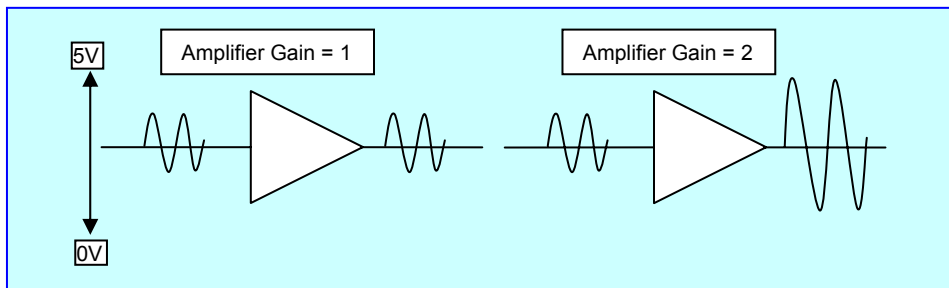
			output), <CR> - carriage return	
7	q – Read bit	/<addr>q<prt><bit><CR>	/ - start char, <addr> - device address, q – command char, <prt> - port number (2), <bit> - bit number (0 to 7), <CR> - carriage return	/Aq25 – for device with address A read port 2 bit 5. The device will return 1 char of bit value (0,1).
8	b – Write bit	/<addr>b<p><v><CR>	/ - start char, <addr> - device address, b – command char, <p> - port number (2), - bit number (0 to 7), <v> - bit value (0,1), <CR> - carriage return	/Ab251 – for device with address A write port 2 bit 5 to 1.

Amplifier Gain Setup

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

Note: Analog signal measurement.

The FLEXEL 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 FLEXEL 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 :

$$Vx = Vref1 + (ADCx - ADC1) * K, \text{ where } K = (Vref2 - Vref1) / (ADC2 - ADC1).$$

Example:

$$Vref1 = 1V; ADC1 = 3200;$$

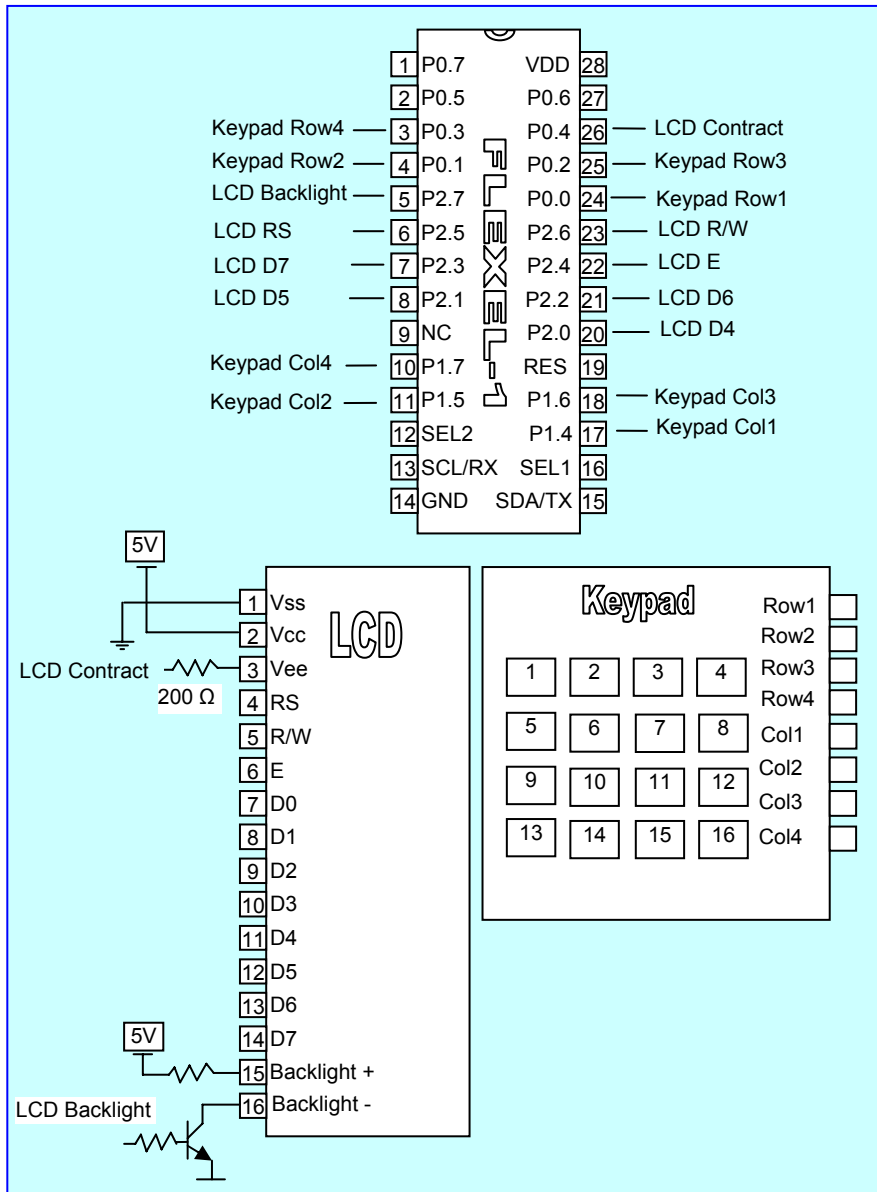
Vref2 = 4V; ADC2 = 12700;

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

3.3 Configuration 3. Character LCD and Keypad control.

The configuration 3 supports the character LCD and 4x4 matrix keypad control.

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



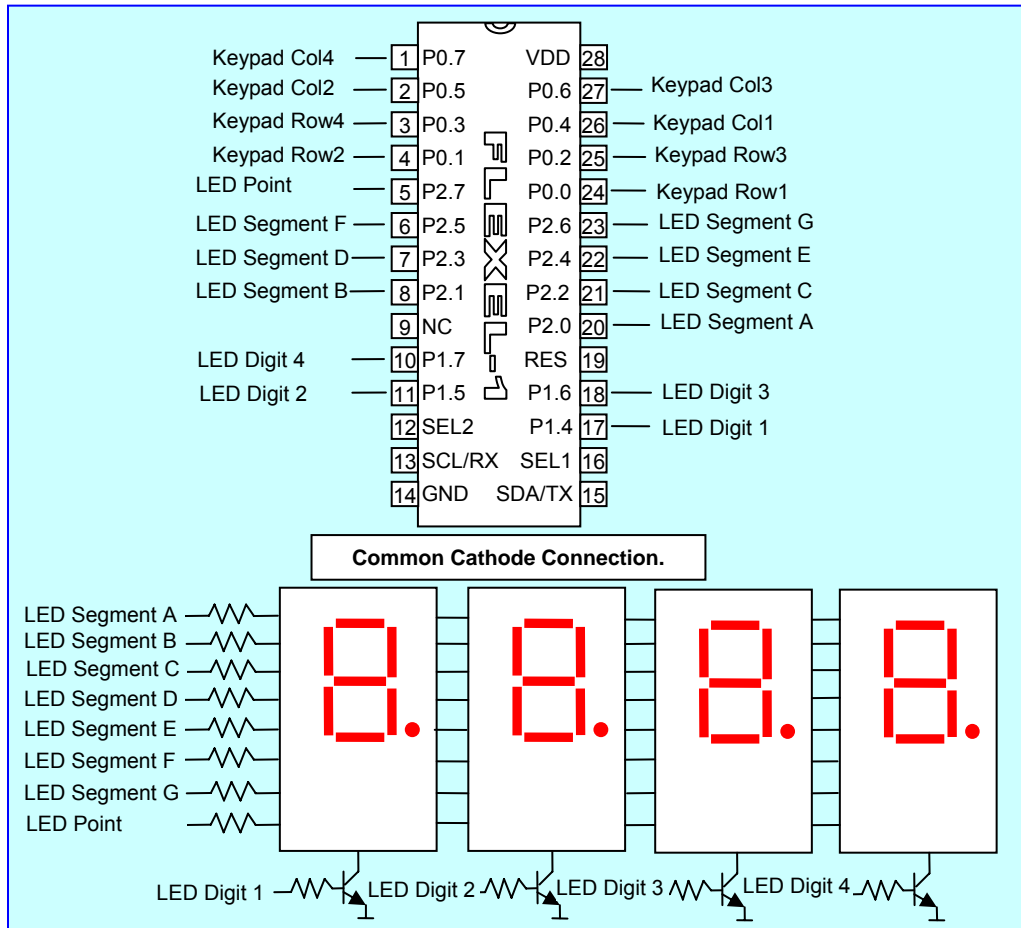
Command Set for Configuration 3.

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 – backlight off, 1- backlight on), <CR> - carriage return	/Ab1 – for device with address A turn on the backlight
6	k – Read the keypad	 /<addr>k<CR>	/ - start char, <addr> - device address, k – command char, <CR> - carriage return	/Ak – for device with address A read the keypad. The device will return the two chars of keypad number (00 – if not button pressed or 01 to 16 button number)

3.4 Configuration 4. LED 7-Segment Display and Keypad control.

The configuration 4 supports Single-Digit, Dual-Digit, Triple-Digit, Quad-Digit Displays and 4x4 matrix Keypad control.

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



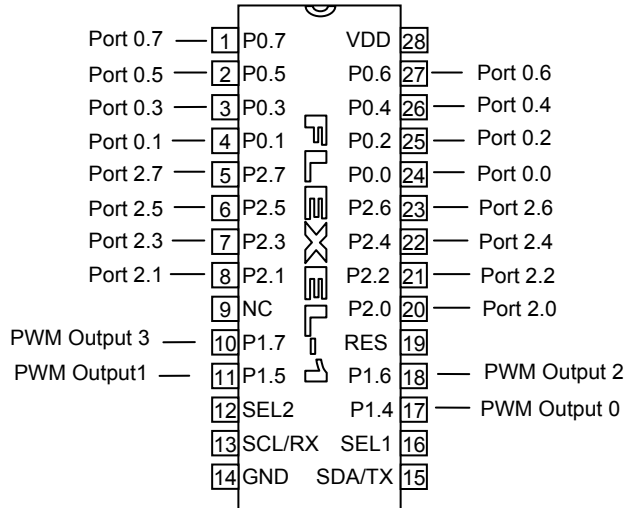
Command Set for Configuration 4.

No.	Command Description	Command Format	Parameters	Example
1	p – Set LED point position	/<addr>p<s><d><CR>	/ - start char, <addr> - device address (ID), p – command char, <s> - enable point 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	/At12342 – for device with address A print string "12.34".
3	k – Read the keypad	/<addr>k<CR>	/ - start char, <addr> - device address, k – command char, <CR> - carriage return	/Ak – for device with address A read the keypad. The device will return the two chars of keypad number (00 – if not button pressed or 01 to 16 button number)

3.5 Configuration 5. DC Motor Control.

The configuration 5 provides four 8 bit PWM blocks to control up to 4 DC motor and 16 digital inputs/outputs for additional tasks.

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



Command Set for Configuration 5.

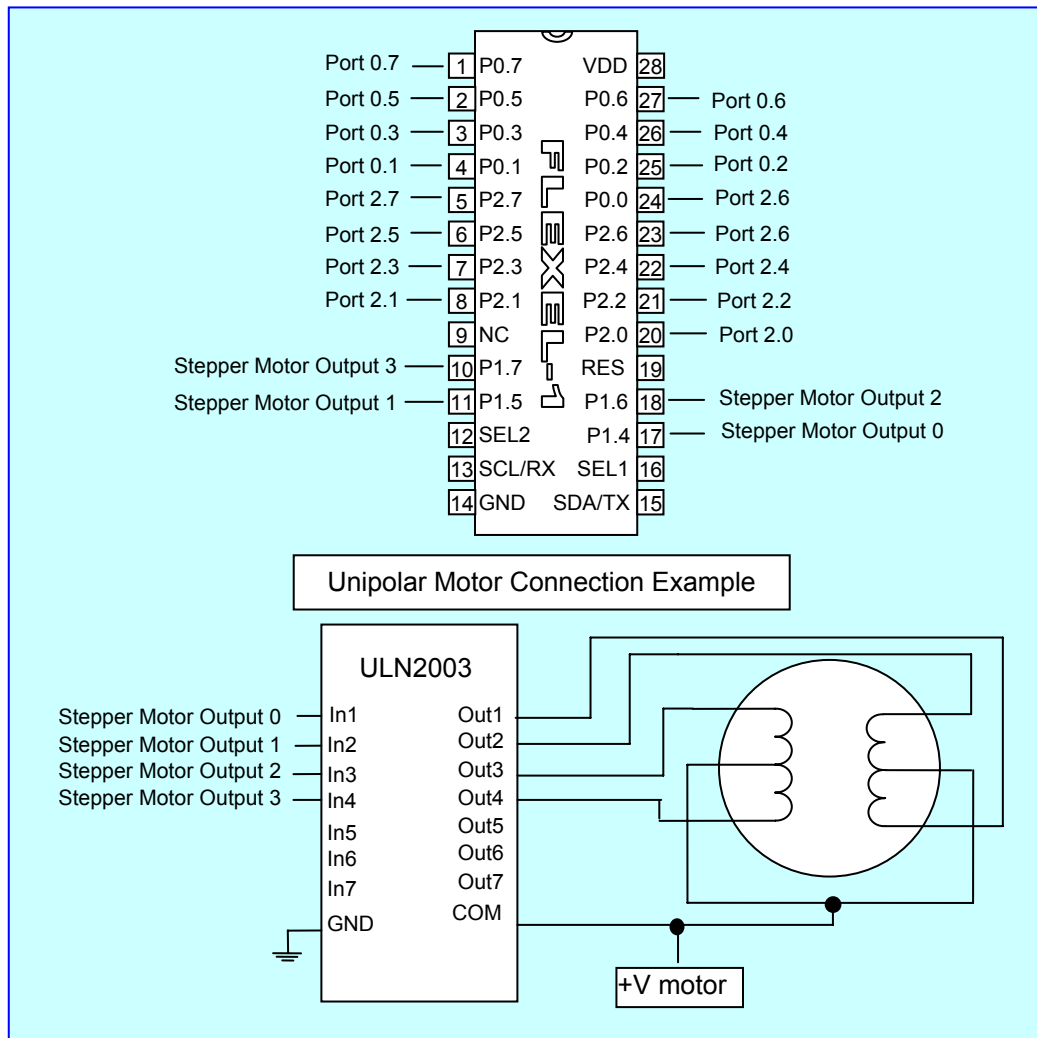
No.	Command Description	Command Format	Parameters	Example
1	p – Set PWM pulse width	/<addr>p<ch><val><CR>	/ - start char, <addr> - device address (ID), p – command char, <ch> - PWM number (0 to 3), <val> - pulse width (0 to 255, 0 – disable PWM), <CR> - carriage return	/Ap2150 – for device A set PWM 2 pulse width to 150.
2	d – Set port direction	/<addr>d<prt><dir><CR>	/ - start char, <addr> - device address, d – command char, <prt> - port number (2), <dir> - direction (0 – input, 1 – output), <CR> - carriage return	/Ad21 – for device with address A set port 2 direction to output
3	r – Read port	/<addr>r<prt><CR>	/ - start char, <addr> - device address, r – command char, <prt> - port number (2), <CR> - carriage return	/Ar2 – for device with address A read port 2. The device will return 3 chars of port value.
4	w – Write to port	/<addr>w<prt><val><CR>	/ - start char, <addr> - device address, w – command char, <prt> - port number (2), <val> - port value, <CR> - carriage return	/Aw2123 – for device with address A write to port 2 value 123
5	u – Set bit direction	/<addr>u<p><d><CR>	/ - start char, <addr> - device address, u – command char, <p> - port number (2), - bit number (0 to 7),	/Au261 – for device with address A set port 2 bit 6 direction to output

			<d> - direction (0 – input, 1 – output), <CR> - carriage return	
6	q – Read bit	/<addr>q<prt><bit><CR>	/ - start char, <addr> - device address, q – command char, <prt> - port number (2), <bit> - bit number (0 to 7), <CR> - carriage return	/Aq25 – for device with address A read port 2 bit 5. The device will return 1 char of bit value (0,1).
7	b – Write bit	/<addr>b<p><v><CR>	/ - start char, <addr> - device address, b – command char, <p> - port number (2), - bit number (0 to 7), <v> - bit value (0,1), <CR> - carriage return	/Ab251 – for device with address A write port 2 bit 5 to 1.

3.6 Configuration 6. Stepper Motor Control.

The configuration 6 provides the stepper motor control and 16 digital inputs/outputs for additional tasks.

Fig.3.7 shows the device pin assignments in 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.
4	r – Read port	/<addr>r<prt><CR>	/ - start char, <addr> - device address, r – command char, <prt> - port number (2), <CR> - carriage return	/Ar2 – for device with address A read port 2. The device will return 3 chars of port value.
5	w – Write to port	/<addr>w<prt><val><CR>	/ - start char, <addr> - device address, w – command char, <prt> - port number (2), <val> - port value, <CR> - carriage return	/Aw2123 – for device with address A write to port 2 value 123
6	u – Set bit direction	/<addr>u<p><d><CR>	/ - start char, <addr> - device address, u – command char, <p> - port number (2), - bit number (0 to 7), <d> - direction (0 – input, 1 – output), <CR> - carriage return	/Au261 – for device with address A set port 2 bit 6 direction to output
7	q – Read bit	/<addr>q<prt><bit><CR>	/ - start char, <addr> - device address, q – command char, <prt> - port number (2), <bit> - bit number (0 to 7), <CR> - carriage return	/Aq25 – for device with address A read port 2 bit 5. The device will return 1 char of bit value (0,1).
8	b – Write bit	/<addr>b<p><v><CR>	/ - start char, <addr> - device address, b – command char, <p> - port number (2), - bit number (0 to 7), <v> - bit value (0,1), <CR> - carriage return	/Ab251 – for device with address A write port 2 bit 5 to 1.

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 Write to Port (send the FLEXEL command '/Aw2157')

```
#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 (FLEXEL address 'A')
slvAddrRD CON $83   ' I2C read address (FLEXEL address 'A')
Darr VAR Byte(8)    ' data array

MAIN:
  PAUSE 100

  Darr(0) = $2F     ' '/' - char
  Darr(1) = $41     ' 'A' - char
  Darr(2) = $77     ' 'w' - char
  Darr(3) = $32     ' '2' - char
  Darr(4) = $31     ' '1' - char
  Darr(5) = $35     ' '5' - char
  Darr(6) = $37     ' '7' - char
  Darr(7) = CR      ' CR' - carriage return

  I2COUT SDA, slvAddrWR, [Darr(0), Darr(1), Darr(2), Darr(3), Darr(4), Darr(5), Darr(6), Darr(7)]

END
```

'An example program Read from Port (send the FLEXEL command '/Ar2')

```
#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 (FLEXEL address 'A')
slvAddrRD CON $83   ' I2C read address (FLEXEL address 'A')
Darr VAR Byte(5)   ' data array

MAIN:
  PAUSE 100

  Darr(0) = $2F    ' '/' - char
  Darr(1) = $41    ' 'A' - char
  Darr(2) = $72    ' 'r' - char
  Darr(3) = $32    ' '2' - char
  Darr(4) = CR     ' CR' - carriage return

  I2COUT SDA, slvAddrWR, [Darr(0), Darr(1), Darr(2), Darr(3), Darr(4)] ' send '/Ar2' command
  PAUSE 1
  I2CIN SDA, slvAddrRD, [STR Darr\3] ' read 3 chars

END
```

'An example program Read analog input (send the FLEXEL 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 (FLEXEL address 'A')
slvAddrRD CON $83   ' I2C read address (FLEXEL 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 '/Ar2'
  PAUSE 15 ' pause 15 ms for ADC data conversion
  I2CIN SDA, slvAddrRD, [STR Darr\5] ' read 5 chars

END
```