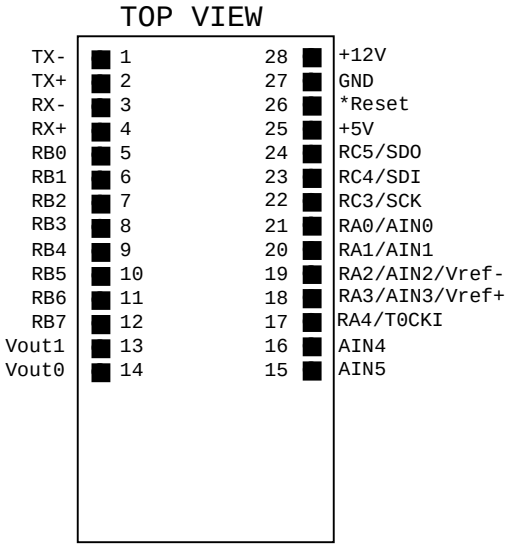
Subminiature Control Computer

Features

- Small size- only 1.5” x 0.875” 28 pin DIP package
- Low power- less than 50mW at 5V
- RS-232A,RS422, and RS485 onboard level shifters
- Hardware USART provides serial communcations from 1.2K to 250Kbps
- SPI bus
- MI^{2}C bus
- BASIC and C compilers are available
- 16 bi-directional, bit-programmable, high current I/O lines: 25-mA sink/source per pin
- 4-channel 10-bit analog-to-digital converter
- 2-channel 12-bit analog-to-digital converter
- 2-channel 12-bit digital-to-analog converter
- Two 8-bit timers and one 16-bit timer
- Up to 14 internal/external interrupt sources
- 8K x 14 words of FLASH program memory
- 368 x 8 bytes of data memory (RAM)
- 256 x 8 bytes of internal data EEPROM
- 1024 x 16 bit external data EEPROM
- In-Circuit, Serial, and Low Voltage Programming are available
- 8MHz system clock
- 500ns instruction cycle



Description

The PicStic 5 is a low cost industrial-oriented controller designed around the PIC16F876. The PicStic 5 incorporates digital inputs, digital outputs, analog inputs, analog outputs, extended data memory, and serial communication all in a single 0.8-cubic-inch encapsulated module.

The PicStic 5 utilizes a pair of communication line drivers to allow several types of serial communication. RS-232A, RS-422, and RS-485 protocols can be used to communicate to the PicStic 5. With an 8MHz-system clock the PicStic 5 has the ability to communicate serially at baud rates up to 250Kbps. This becomes beneficial when time is a factor.

SPI and I^{2}C communications are also available. These buses are made accessible through pins 23-25. During I^{2}C communication the SCL frequency can be set at 100KHz, 400KHz, or 1MHz. During SPI and I^{2}C communications, bits can be clocked out at rates up to 2MHz. In SPI and I^{2}C modes received data is double buffered allowing a second byte of data to be received before the first byte is processed.

Accompanying the 4-channels of 10-bit analog-to-digital converters supplied by the PIC16F876 processor, the PicStic 5 also offers 2-channels of 12-bit analog-to-digital conversion. The two 12-bit channels can be sampled at approximately 10K samples/sec. The four 10-bit ADC channels can be sampled at approximately 20K samples/sec.

There are 2-channels of 12-bit digital-to-analog conversion included in the PicStic 5. These analog outputs have a range from 0-4.095V. Both channels have the ability to be loaded simultaneously.

The PicStic 5 has 16 bi-directional bit-programmable, high-current I/O lines capable of sinking/sourcing 25mA.

Absolute Maximum Ratings

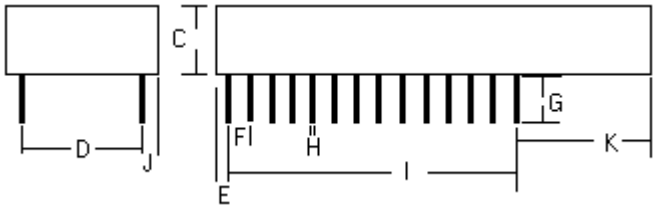
Supply Voltage	+16 V
Digital Input Voltage	-0.5 V to +6.5 V
Analog Input Voltage	-0.5 V to +5.5 V
Lead Temperature	260°C
Storage Temperature	-25°C to +100°C
Operating Temperature	0°C to +70°C
Industrial Temperature	-40°C to +100°C

(Industrial versions are available by special order)

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings if this specification in not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Mechanical Specifications

	Inches		Millimeters	
Dim	Min	Max	Min	Max
A	2.19	2.21	55.62	56.13
B	0.86	0.88	21.84	22.35
C	0.364	0.384	9.25	9.75
D	0.59	0.61	14.98	15.49
E	0.085	0.22	2.16	5.59
F	0.095	0.105	2.41	2.66
G	0.19	0.21	4.83	5.33
H	0.02	0.03	0.51	0.76
I	1.29	1.32	32.76	33.53
J	0.075	0.185	1.9	4.69
K	0.74	0.77	18.79	19.55



Pin Description

The PicStic 5 is a 28-pin DIP with 0.6" pin spacing.
A pin that has multiple signal designations listed indicates the default and optional functions for that pin. To use optional pin functions, the appropriate registers in the processor corresponding to those functions need to be configured properly.

Pin	Signal	Description
1	TX-	RS-422/-485/-232A inverted serial output line (transmit pair/ rec-xmit pair)
2	TX+	RS-422/-485 noninverted serial output line (transmit pair/rec-xmit pair/transmit)
3	RX-	RS-422/-485/-232A inverted serial input line (receive pair/ rec-xmit pair)
4	RX+	RS-422/-485 noninverted serial input line (receive pair/rec-xmit pair/receive)
5-12	RB0-RB7	Processor general-purpose TTL-level programmable input/ ouput pins. Each pin can sink and source 25mA of current. Total current for this port should not exceed 50mA (RB0 can also be used as an external interrupt, RB3, RB6, and RB7 are used for low voltage serial programming
13	Vout1	12-bit analog output channel 1, 0-5V range
14	Vout0	12-bit analog output channel 0, 0-5V range
15	Ain5	12-bit analog input channel 5, 0-5V range
16	Ain4	12-bit analog input channel 4, 0-5V range
17	RA4/ T0CKI	Processor general-purpose TTL-level programmable input/ouput pins. RA4 is also clock input for Timer0
18-19	Ain3/RA3/Vref+ Ain2/RA2/ Ref-	Processor general-purpose TTL-level programmable input/output. Can source/ sink 25mA. These pins can also be configured as the 10-bit ADC channels 3 and 2. The Vref+ and Vref- are also located on these pins.
20-21	Ain1/RA1 Ain0/RA0	Processor general-purpose TTL-level programmable input/output. Can source/ sink 25mA. These pins can also be configured as the 10-bit ADC channels 1 and 0
22-24	RC3/SCK- RC4/SDI- RC5/SDO	General purpose I/O. Can sink source 25mA. These pins are also the SPI and I^2C buses
25	+5V	This is the +5V internal ADC and DAC voltage reference (nominally 5.0V). This pin may be used to power minimal external circuitry or sensors. The PicStic 5 may be powered by +5V only through this pin provided pin 28 is left unconnected.
26	RESET	When grounded, RESET provides a master clear of the system. All outputs are cleared
27	GND	Combined digital and analog system ground
28	V+	PicStic 5 power supply input V+ is nominally 8-12 volts If pin 28 is left open then the PicStic 5 can be powered by +5V through pin 25

DC Electrical Characteristics

Operating Temperature T_a = 0°C to 70°C
Operating Voltage V_{RAW} = 12 V GND = 0 V

Characteristics	Min	Typical	Max	Units	Condition
Supply Voltage (V_{RAW})	9	12	16	V	
Supply Current (I_{cc})		4		mA	
Input Low Voltage (V_{il})	-0.5		0.9	V	
Input High Voltage (V_{ih})	1.9		5.5	V	
Output Low Voltage (V_{ol})		0.45		V	I_{ol} = 1.6 mA
Output High Voltage (V_{oh})	4.5			V	I_{ol} = -10 µA
	2.4			V	I_{ol} = -400µA

Communication Line DC Electrical Characteristics

Characteristics	Min	Typical	Max	Units	Condition
Differential Driver Output Voltage			5.0	V	Unloaded; See note 1
RS-422	2		5.0	V	R = 50 Ω
RS-485	1.5		5.0	V	R = 25 Ω
Maximum Receiver Input Voltage	-14		14	V	
ESD Protection		2000		V	

D/A Converter Characteristics

Characteristics	Min	Typical	Max	Units	Condition
Resolution	12			bit	
Full-Scale Error		1/2		bit	
Offset Error		± 2.0	± 8.0	mV	
Voltage Reference		5		V	V_{ref} is V_{cc}
Analog Output Impedance		150	300	Ω	measured to GND

A/D Converter Characteristics (10-bit)

Characteristics	Min	Typical	Max	Units	Condition
Resolution	10			bit	
Linearity Error		<±1		LSb	See Note 4
Offset and Gain Error		<±1		LSb	See Note 4
Voltage Reference (V_{ref+} - V_{ref-})	2.0		V_{cc} + 0.3	V	See Note 3
Voltage Reference High (V_{ref+})	V_{cc} - 2.5		V_{cc} + 0.3	V	See Note 3
Voltage Reference Low (V_{ref-})	V_{ss}-0.3		V_{ref+} - 2.0	V	See Note 3
Analog Input Range	V_{ss}-0.3		V_{ref} + 0.3	V	See Note 3
Analog Input Impedance		10K		Ω	
Sampling Rate			20.8K	smp/sec	See Note 5

A/D Converter Characteristics (12-bit)

Characteristics	Min	Typical	Max	Units	Condition
Resolution	12			bit	
Linearity Error		3/4		bit	
Offset and Gain Error		2.0		bit	
Voltage Reference	4.5	5.0	5.5	V	V_{ref} is V_{cc}
Analog Input Range		-0.5 to V_{ref} + 0.5		V	See note 2
Analog Input Impedance		250K		Ω	
Sampling Rate			~10K	smp\sec	

Note 1: RS-232A is characterized as a +/-5V bipolar signal (as opposed to RS-232C at +/-12V). Drivers and receivers are actually RS422 and the serial interface is an RS-423 connection (single ended to differential).

Note 2: Two diodes are tied to each analog input which will conduct when the input voltage is one voltage drop below Vss or one voltage drop above V_{ref}/V_{cc}. To achieve absolute 0-5 V input range requires Vref to be greater than 4.950 V.

Note 3: When using an external V_{ref} only two channels of 10-bit A/D conversion are available. The external V_{ref+} must be applied to Pin 18 and the external V_{ref-} must be applied to Pin 19. Each external reference can be disabled (individually or simultaneously) by setting the appropriate bit in the ADCON1 register. When external V_{ref+} is disabled, V_{cc} then becomes the positive reference. When external V_{ref-} is disabled, V_{ss} then becomes the negative reference.

Note 4: Values for integral linearity, differential linearity, gain, and offset errors are based on V_{ref} = V_{cc} = 5.12V and $V_{ss} \le V_{ain} \le V_{ref}$.

Note 5: This is the MAX. sample rate per second. This rate can only be achieved by configuring bits 6 and 7 correctly in the ADCON0 register. Bit 6 is ADCS0 and must be configured to a '0'. Bit 7 is ADCS1 and must be configured to a '1'

1.0 Quick Start

You will probably want to hook up your new PicStic 5 and try it out without having to read this entire datasheet. Use the following steps to connect and configure the PicStic 5 with a simple BASIC program. Be sure to thoroughly read the remaining sections of this manual before trying to use any of the features not described here.

Be sure that PicBasic Pro Compiler and Epic software are already installed on your PC's hard drive. The PicBasic Pro Compiler needs to be version 2.11 or higher and the Epic programmer needs to be version 2.10 or higher to program the PicStic 5. Using any text editing program type the program listed below and save it in an ASCII text only format, with a .BAS extension. The program should be saved in the PBP directory. The program will need to be compiled from a DOS prompt. The command line will look like this:

C:\pbp>pbp –pps5 quick.bas

The .HEX file will automatically be saved back into the PBP directory.

Plug the PicStic Programmer 2 board into the parallel port of your PC. Apply power to the board. Make certain that all jumpers on JP2 of the programmer are configured in the DIP position. From the Epic directory run the EpicWin application. When the program starts, select PIC16F876 from the dropdown list of processors. Open the file called Quick.hex to be loaded. On the EpicWin menu bar click View and then Configuration. Oscillator should be set to HS, code protect should be set to OFF, and power-up timer should be enabled.

Now your ready to place the PicStic 5 into the programmer board, be sure that the module is upper-justified in the ZIF socket. Click the erase button to clear the program memory and then click the program button to load the new program. Now you can place your programmed PicStic into your application. For this application you will need to apply 9-12VDC to pin 28 (or a regulated 5VDC to pin 25) and GND to pin 27. **NOTE: Applying power backwards or to the wrong pins will certainly damage the PicStic (and void the warranty).** An oscilloscope or logic probe will show all 16 I/O counting in a binary fashion. If you do not have access to one of the instruments mentioned, an LED with a series resistor can be used to see any output bit toggling.

```
COUNTERA      VAR     BYTE                  'Define the variables to be used
COUNTERB      VAR     BYTE
COUNTERC      VAR     BYTE

TRISA = %11100000                   'Set PortA bits 0-4 as outputs with binary value
TRISB = 0                           'Set all of PortB as outputs with a decimal value
TRISC = $0C7                        'Set bits 3-5 of PortC as outputs with a hex value

START:        FOR COUNTERC = 7 TO 63 STEP 8 'Set counter to toggle bits 3-5
              CALL LOOP2
              PORTC = COUNTERC              'Sets PortC equal to counter
              NEXT COUNTERC                 'Increment the count
              GOTO START                    'Lets do it again
LOOP2:        FOR COUNTERA = 0 TO 31        'Sets counter to toggle bits 0-4
              CALL LOOP1
              PORTA = COUNTERA              'Sets PortC equal to counter
              NEXT COUNTERA                 'Increment the count
              RETURN
LOOP1:        FOR COUNTERB = 0 TO 255       'Sets PortB to do a full 8-bit count
              PORTB = COUNTERB              'Set PortB equal to the counter
              NEXT COUNTERB                 'Increment the count
              RETURN
              END                           'End of the program
```

Figure 1.0: This Quick start program performs a binary count of all 16 I/O pins.

2.0 PicStic 5 Software

When it comes from the factory, the PicStic 5 has the factory test program in the user program space. Code is developed using cross-development tools running on a desktop PC and programmed into the PicStic 5 for execution. There are several development environments from which to choose.

2.1 Assembly Language

Any cross-assembler capable of creating code for the PIC16F876 processor can be used to write assembly language programs for the PicStic 5. Microchip's assembler is available at no cost from their Web site. More information about the PIC instruction set may be found in the Microchip Data Book. Their Web site may be contacted at http://www.microchip.com/.

Other cross-assemblers are available including one from various vendors that enhance the PIC instruction set with one more familiar to 8051 programmers.

A cross-assembler also comes with the Micromint PicBasic Pro development package. Assembly language routines to access the PicStic 5 peripherals are available upon request.

2.2 C Compiler

While the PicStic 5 documentation centers on the use of the PicBasic Pro compiler, a PIC C cross-compiler is available. The integrated C development environment gives developers the capability to quickly produce efficient code from an easily maintainable high-level language. The compiler includes built-in functions to access the PIC hardware such as INPUT and OUTPUT_HIGH. Variables including structures may be directly mapped to memory such as I/O ports to best represent the hardware structure in C.

Functions may be implemented inline or separate. Function parameters are passed in reusable registers. Inline functions with reference parameters are implemented efficiently with no overhead. While a C compiler is available, Micromint does not currently provide support for it. Assembly language routines to access the PicStic 5 peripherals are available upon request.

2.3 PicBasic Pro

The PicBasic Pro compiler is an English-like language, which makes it much easier to read and write than Microchips assembly language. For operation of the PicBasic Pro compiler you will need a text editor or word processor for developing your program source file. The EPIC programmer is also available for easy programming of the PicStic 5. Programming the PicStic 5 will be covered in section 3.1. The instruction set of the PicBasic Pro compiler has a multitude of commands that will make code development for the PicStic 5 a relatively painless experience. READ and WRITE commands allow developers to utilize the internal data EEPROM with a single instruction. I^2C instructions allow externally added I^2C peripherals to be accessed with a single command. If time is essential, assembly code can be added directly into your PicBasic Pro program using the ASM..ENDASM commands. PicBasic Pro also provides direct access to the direction control registers (TRIS) of all I/O Ports in the PIC16F876. These are just a few of the benefits that the PicBasic Pro compiler offers the PicStic 5. For a complete list of PicBasic Pro instructions refer to Section 2.4. The PicBasic Pro compiler automatically includes routines to access the internal peripheral of the PicStic 5.

To purchase the latest version of the PicBasic Pro Compiler please contact the Micromint Sales Department.

2.4 PicBasic Pro (version 2.33) Instruction Set

@ - Insert one line of assembly language code.
ADCIN - Read on-chip analog to digital converter.
ASM..ENDASM - Insert assembly code section.
BRANCH - Computed GOTO
BRANCHL - BRANCH out of page
BUTTON - Debounce &auto-repeat input on pin.
CALL - Call assembly language subroutine.
CLEAR - Zero all variables.

CLEARWDT - Clear Watchdog Timer.
COUNT - Count number of pulses on a pin.
DATA - Define initial contents of on-chip EEPROM.
DEBUG - Asynchronous serial output to fixed pin and baud.
DEBUGIN - Asynchronous serial input from fixed pin and baud.
DISABLE - Disable ON INTERRUPT and ON DEBUG processing.
DISABLE DEBUG - Disable ON DEBUG processing.
DISABLE INTERRUPT - Disable ON INTERRUPT processing.
DTMFOUT - Produce touch-tones on a pin.
EEPROM - Define initial contents of on-chip EEPROM.
ENABLE - Enable ON INTERRUPT and ON DEBUG processing.
ENABLE DEBUG - Enable ON DEBUG processing.
ENABLE INTERRUPT - Enable ON INTERRUPT processing.
END - Stop execution and enter low power mode.
FOR..NEXT - Repeatedly execute statements.
FREQOUT - Produce up to 2 frequencies on a pin.
GOSUB - Call BASIC subroutine at specified label.
GOTO - Continue execution at specified label.
HIGH - Make pin output high.
HSERIN - Hardware asynchronous serial input.
HSEROUT - Hardware asynchronous serial output.
I2CREAD - Read bytes from I2C device.
I2CWRITE - Write bytes to I2C device.
IF..THEN..ELSE..ENDIF - Conditionally execute statements.
INPUT - Make pin an input.
{LET} - Assign result of an expression to a variable.
LCDIN - Read RAM on LCD.
LCDOUT - Display characters on LCD.
LOOKDOWN - Search constant table for value.
LOOKDOWN2 - Search constant / variable table for value.
LOOKUP - Fetch constant value from table.
LOOKUP2 - Fetch constant / variable value from table.
LOW - Make pin output low.
NAP - Power down processor for short period of time.
ON DEBUG - Execute BASIC debug monitor.
ON INTERRUPT - Execute BASIC subroutine on an interrupt.
OUTPUT - Make pin an output.
PAUSE - Delay (1mSec resolution).
PAUSEUS - Delay (1uSec resolution).
PEEK - Read byte from register.
POKE - Write byte to register.
POT - Read potentiometer on specified pin.
PULSIN - Measure pulse width on a pin.
PULSOUT - Generate a pulse on a pin.
PWM - Output pulse width modulated pulse train to pin.
RANDOM - Generate pseudo-random number.
RCTIME - Measure pulse width on a pin.
READ - Read byte from on-chip EEPROM.
RESUME - Continue execution after interrupt handling.
RETURN - Continue execution at statement following last executed GOSUB.
REVERSE - Make output pin an input or an input pin an output.
SERIN - Asynchronous serial input (8N1) (BS1 style with timeout.)
SERIN2 - Asynchronous serial input (BS2 style.)
SEROUT - Asynchronous serial output (8N1) (BS1 style.)
SEROUT2 - Asynchronous serial output (BS2 style.)
SHIFTIN - Synchronous serial input.
SHIFTOUT - Synchronous serial output.
SLEEP - Power down processor for a period of time.
SOUND - Generate tone or white-noise on specified pin.
STOP - Stop program execution.
SWAP - Exchange the values of two variables.
TOGGLE - Make pin output and toggle state.
WHILE..WEND - Execute code while condition is true.
WRITE - Write byte to on-chip EEPROM.
XIN - X-10 input.
XOUT - X-10 output.

Functions / Operators:

ABS - Absolute value
COS - Cosine
DCD - 2n decode
DIG - Return digit
MAX - Maximum
MIN - Minimum
NCD - Encode
REV - Reverse bits
SIN - Sine
SQR - Square root

3.0 Programming PicStic 5

The user-programmable portion of the PicStic 5 uses Microchip Technology's PIC16F876 EEPROM microcontroller, which can be reprogrammed hundreds of times. These programs can be created using a number of resources, as described in Section 2.

Programming the PicStic 5 is done serially, involving only five connections. The five signals are power (+5V on pin 25), ground (GND on pin 27), *RESET (pin 26) must be pulled to +12V, and port pins RB6 (serial clock) and RB7 (serial data).

The PicStic PicBasic Pro Development package includes a programmer, which is plug-compatible with the PicStic 5 and needs no programming adapters. The PicStic 5 can be programmed with most other PIC16F876 programmers by making a simple five-wire adapter to connect these five signals appropriately. Most programmers and compilers enable you to take advantage of the configuration options. Some of these options include code protection, power-up timer, watchdog timer, low voltage programming and oscillator designation. Be aware that you may have to set these conditions on the programmer menu for proper programming of the PicStic 5.

For a more detailed description of the PIC16F876 programming algorithm and technology, refer to the *Microchip Data Book.*

3.0 Serial Communications

The PicStic 5 offers several methods of communication, allowing maximum flexibility in setting up the target application. The options include RS-232A, RS-422, and RS-485.

3.1 RS-232A

The primary serial interface on the PicStic 5 is an inverted RS-232A interface. When people talk about RS-232, they are usually referring to RS-232C. RS-232C uses high, bi-polar voltages (typically ±12V) to send serial data. RS-232A is a low-voltage version of the same signal. RS-232C and RS-232A can often co-exist since interface chips used in both can usually handle a wide range of voltages.

By connecting the TX- and RX- lines of the PicStic 5 to a normal RS-232C port, the module can be used by most any device with a serial port. The receiver on the PicStic 5 can handle the high voltages used by RS-232C and most RS-232C receivers have low enough switching thresholds that they can easily detect the 0-5V swings of the PicStic 5's transmitter.

3.2 RS-422

RS-422 uses differential signaling to achieve reliable communication distances of up to 4000 feet (versus 50 feet with RS-232C). Similar to RS-232, RS-422 allows you to connect just two devices together. A 100Ω terminating resistor should be placed across each pair at the receiving end of any long cable runs. It is up to the user to provide this termination.

3.3 RS-485

RS-485 is very similar to RS-422 but uses just one twisted pair of wires. Although this arrangement allows only one device to transmit at a time, it allows up to 32 devices to be connected to the same wire pair. Multiple PicStic 5 modules may be connected to a master to form a network. When more than 32 devices are required, Micromint RS-485 repeaters may be used to break the 32-device barrier.

A 100Ω terminating resistor should be placed across the wire pair at each end of the wire. There should not be more than two such resistors on any network. To improve reliability, it is also a good idea to include a 1KΩ pull-up resistor on the positive side and a 1KΩ pull-down resistor on the negative side of the wire pair. Only one set of pull-up and pull-down resistors should be used on any network. It is up to the user to provide all terminating and pull-up/pull-down resistors.

4.0 Analog-to-Digital Converters

The PicStic 5 is equipped with a total of six channels of analog-to-digital conversion. Four 10-bit channels are provided by the PIC16F876 processor and two 12-bit channels are available though an LTC1298.

4.1 10-Bit (PIC16F876)

The four 10-bit channels of A/D are available on pins 18-21. Pins 18-21 are multi-function pins. They can be configured as Digital I/O or 10-bit analog inputs. The 10-bit analog inputs are enabled by default when power is applied to the module. The ADCON1 register in the PIC16F876 processor controls the functions of these pins. See Figure 4.0 and 4.1 for configuring the A/D channels.

The ADCON0 register shown in Figure 4.0 controls the operation of the A/D module. The ADCON1 register shown in Figure 4.1 configures the functions of the port pins. **NOTE:** Configuring ADCON1 in a manner different than described in Fig 4.1 could cause the PicStic 5 not to operate properly. While the purpose of this manual is to provide a technical description of the PicStic 5 and it's capabilities, it is beyond the scope of this manual to provide a detailed description of the inner workings of the PIC16F876 processor. For a more detailed description of these registers, refer to Microchip Technology's PIC16F876 data sheet.

The following steps should be followed for correctly doing an A/D conversion:

1. Configure the A/D module:
   - Configure ADCON1 (analog pins, voltage reference, and digital I/O)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit
3. Wait the required acquisition time
4. Start the conversion:
   - Set GO/*DONE bit (ADCON0)
5. Wait for A/D conversion to complete by either:
   - Waiting for the A/D interrupt

OR

   - Polling for the GO/*DONE bit to be cleared.
6. Read A/D result register pair (ADRESH {located at 1EH of Bank 0}: ADRESL {located at 9EH of Bank 1}), clear bit ADIF if required.
7. For next conversion, go to step 1 or 2 as required. The A/D conversion time per bit is defined as T_{ad}. A minimum wait of 2Tad is required before the next acquisition starts.

Referenced from the PIC16F87x Data Sheet

When the PicStic 5 is configured correctly the 10-bit A/D channels can reach sampling rates of approximately 20K samples per second.

4.2 12-Bit (LTC1298)

Sometimes 10-bit resolution is not enough, so two channels of 12-bit A/D have been included in the PicStic 5. In an attempt to make the PicStic 5 easy to use, a custom software routine has been written (in PicBasic Pro) to access the LTC1298. This routine is automatically included when the user program is compiled and handles all communications with the LTC1298.

The included code is available after the code is compiled. After the file has been included all that needs to be done is to execute a CALL to the subroutine. The conversion will be returned back into a designated variable (PS5ADC).

The routine was written to allow a single-ended read of either channel or a differential read of the two channels. NOTE: Only one channel can be read at a time. Different CALLs were set up to designate single-ended reads and differential reads. Regardless of which sampling option is chosen, the conversion will always be stored into the PS5ADC variable. PS5ADC is a predefined variable in the LTC1298 Include file, thus reserving it for the A/D CALLs. The sample program in Figure 4.2 illustrates how to read the 12-bit A/D channels using PicBasic Pro.

There is four CALL commands that are available with the LTC1298 Include file. 'CALL AIN5' takes a single-ended read of channel 0 on the LTC1298. 'CALL AIN6' takes a single-ended read of channel 1 of the LTC1298. 'CALL AD01' takes a differential read of Ch0-Ch1. 'CALL AD10' takes a differential read of Ch1-Ch0. After each read the value returned in PS5ADC needs to be processed or saved into a different variable or the data will be lost.

ADCON0 REGISTER (ADDRESS 01Fh)

ADCS1	ADCS0	N/A**	CHS1	CHS0	GO/*DONE	N/A**	ADON
bit7							bit0

bit 7-6: **ADCS1:ADCS0:** A/D Conversion Clock Select bits
00 = F_{osc} /2
01 = F_{osc} /8
10 = F_{osc} /32
11 = F_{RC} (clock derived from an RC oscillation)

bit 5: Not available**

bit 4-3: **CHS1:CHS0:** Analog Channel Select bit
00 = AIN0
01 = AIN1
10 = AIN2
11 = AIN3

bit 2: **GO/*DONE:** A/D Conversion Status bit
If ADON = 1
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (When conversion is complete this bit is cleared)

bit 1: Not available**

bit 0: **ADON:** A/D On bit
1 = A/D converter module is operating
0 = A/D converter is in shutoff

****NOTE:** These bits are to be read as '0'

Referenced from the PIC16F87x Data Sheet

Figure 4.0: ADCON0 Register

ADCON1 REGISTER (ADDRESS 09Fh)

ADFM	N/A**	N/A**	N/A**	PCFG3	PCFG2	PCFG1	PCFG0
bit7							bit0

bit 7 **ADFM:** A/D Result Format Select
1 = Right Justified. 6 most significant bits of ADRESH are read as '0'
0 = Left Justified. 6 least significant bits of ADRESL are read as '0'

bit 6-4 N/A**

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits

PCFG3: PCFG0	RA3 AIN3	RA2 AIN2	RA1 AIN1	RA0 AIN0	V_{ref+}	V_{ref-}
0000	A	A	A	A	V_{CC}	V_{SS}
0001	V_{ref+}	A	A	A	RA3	V_{SS}
0100	A	D	A	A	V_{CC}	V_{SS}
0101	V_{ref+}	D	A	A	RA3	V_{SS}
0111	D	D	D	D	V_{CC}	V_{SS}
1000	V_{ref+}	V_{ref-}	A	A	RA3	RA2
1111	V_{ref+}	V_{ref-}	D	A	RA3	RA2

A = Analog input
B = Digital I/O

****NOTE:** These bits are to be read as '0'

Referenced from the PIC16F87x Data Sheet

Figure 4.1: ADCON1 Register

```
Define  HSER_BAUD       2400                    'Set the baud rate to transmit at
Define  HSER_RCSTA      90H                     'Config Receive Control Register
Define  HSER_TXSTA      20H                     'Config Transmit Control Register


W   VAR  BYTE                                   'program to read the A/D

START:    HIGH PORTC.1                          'Enable the transmitter
          PAUSE 100                             'Tenth sec pause to enable trans.
          HSEROUT ["Verifying that the ADC call works.",13,10]
          HSEROUT ["Press any key to continue.",13,10]
          HSERIN  [w]                           'Wait for user to confirmation
          PAUSE 1000                            'One second pause
          CALL AIN5                             'Sample channel 0 of the LTC1298
          HSEROUT ["Channel 0 of the 1298 reads ", dec PS5ADC, 13,10] 'Print results
          PAUSE 1000                            'Print results serially
          CALL AIN6                             'Sample channel 1 of LTC1298
          HSEROUT ["Channel 1 of the 1298 reads ", dec PS5ADC, 13,10]
          PAUSE 1000                            'Print results serially
          CALL AD01                             'DIFF read of CH0 to CH1
          HSEROUT ["Channel DIF of the 1298 reads ", dec PS5ADC, 13,10]
          PAUSE 1000                            'Print results serially
          CALL AD10                             'DIFF read of CH1 to CH0
          HSEROUT ["Channel DIF of the 1298 reads ", dec PS5ADC, 13,10]
          PAUSE 2000                            'Print results serially
          END
```

Figure 4.2: Example program demonstrating how to use the 'LTC1298' using PicBasic Pro.

The two 12-bit ADC channels can be sampled at approximately 10K samples per second. This sample rate was derived using the PicBasic Pro Compiler and the provided Include file. Sample rates may differ higher or lower if straight assembly language or a C compiler is used. The Include file was written to simplify the use of the PicStic 5 as well as to optimize the abilities of the LTC1298. While the maximum sampling rate of the LTC1298 is 11.1K, samples taken at the maximum rate are not recommended or guaranteed.

5.0 Digital-to-Analog Converter (LTC1446)

Included in the PicStic 5 is an LTC1446, which provides two channels of 12-bit D/A. Like the 12-bit A/D; the D/A also has a custom Include file to make using the PicStic 5 easier. The Included file takes care of all communications between the PIC16F876 and the LTC1446. The routine is automatically included when the user program is compiled. To use the routine simply load the 12-bit value(s) to be converted into the appropriate variable(s), and then CALL 'DACOUT'. The variables used (PS5DAC(0) ⇒ channel 0 of the LTC1446, PS5DAC(1) ⇒ channel 1 of the LTC1446) are predefined, thus reserving them for the 'DACOUT' routine.

Both 12-bit channels are loaded and updated simultaneously. When only 1 channel is going to be used the other should be loaded with a zero value. The output range of the D/A is 0 to 4.095V. Figure 5.1 demonstrates how to use the 'LTC1446' Include file.

When using the 12-bit analog outputs, it may be necessary to power the PicStic 5 through the external 5V pin. The LTC1446 requires a lot of current when under a heavy load, sometimes more than the PicStic's internal regulator can provide. If an external 5V is required, a **regulated** 5V can be applied to pin 25 of the PicStic module.

The LTC1446 has an internal reference and a full-scale output of 4.095V. Refer to Fig. 5.0 for a sample program illustrating how to access the LTC1446 using PicBasic Pro.

```
DEFINE  HSER_BAUD     2400                  'Set baud rate
DEFINE  HSER_RCSTA     90H                  'Config Receive Control Register
DEFINE  HSER_TXSTA     20H                  'Config Transmit Control Register


START:  HIGH PORTC.1                        'Enable Transmitter
        PAUSE 100
         HSEROUT ["Enter a 3 digit HEX value for PS5DAC(0).",13,10]
         HSERIN  [hex3 PS5DAC(0)]           'Input a 12-bit value
         HSEROUT ["Enter a 3 digit HEX value for PS5DAC(1).",13,10]
        HSERIN  [hex3 PS5DAC(1)]            'Input a 12-bit value
         HSEROUT  [hex PS5DAC(0),10,13, hex PS5DAC(1),10,13]
         CALL DACOUT                        'Load both D/A channels
         PAUSE 1000                         'One second pause
         END
```

Figure 5.0: Example program demonstrating how to use the 'LTC1446' using PicBasic Pro.

6.0 Serial EEPROM (93LC86)

A 16Kbit x 16 serial EEPROM was added to the PicStic 5 for simple data logging applications. Communication routines between the PIC16F876 and the 93LC86 have been written to allow easy data storage. The PIC16F876 uses the SPI bus to communicate with the 93LC86, allowing transmission rates of up to 2MHz. Similar to the A/D and D/A routines, the EEPROM routine will be included when the user program is compiled. Two variables have been predefined, one for the address (PS5EEADD), and one for the data (PS5EEDAT).

The serial routines were written to simplify the learning curve of using the PicStic 5. There are separate CALL routines in the 93LC86 Include file. One call writes to the EEPROM (EEWRITE), and one call reads from the EEPROM (EEREAD).

When using the EEWRITE routine the address being written to needs to be stored into the predefined variable PS5EEADD. The data needs to be stored into the predefined data variable, PS5EEDAT. Once the data and address are properly stored the CALL EEWRITE routine can be executed. The EEREAD routine works similar to the EEWRITE routine. Load the address to be read into the predefined variable then CALL EEREAD. The value read will be returned into the data variable. The contents of the PS5EEDAT before the EEREAD routine is called is not significant, the value read will be overwritten. Figure 6.0 illustrates how to use the EEPROM CALL routine.

```
Define  HSER_BAUD          2400                 'Set the baud rate to transmit at
Define   HSER_RCSTA        90H                          'Config Receive Control Register
Define   HSER_TXSTA        20H                          'Config Transmit Control Register

STORE   VAR         WORD

START:      HIGH PORTC.1                        'Enable the transmitter
            HSEROUT ["ENTER NUMBER BETWEEN 0 - 65535 FOR THE DATA" ,10,13]
            HSERIN [DEC5 SEEDATA]               'Receive a 5 digit decimal number for data
            HSEROUT ["ENTER NUMBER BETWEEN 0 - 1024 FOR THE ADDRESS" ,10,13]
            HSERIN [DEC4 SEEADDR]               'Receive a 4 digit decimal number for Addr
            CALL EEWRITE                        'Go write to the EEPROM
            SEEDATA = 00                        'Set the data variable to zero
            CALL EEREAD                         'Retrieve the data previously wrote
            HSEROUT ["HERE IS THE VALUE STORED "]
            HSEROUT [DEC SEEDATA,10,13,10,13]   'Display the data read
            END
```

Figure 6.0: Example program demonstrating how to use the '93LC86' using PicBasic Pro.

7.0 CALL Routines and Variables

The PicStic 5 has routines written to access all internal peripherals. These routines were written to be included as part of the PicBasic Pro Compiler. The assembly language routine can be supplied separately (upon request and with proof of a PicStic 5 purchase) and with minor modifications the routines will function properly with C Compilers and most assemblers.

Each peripheral internal to the PicStic 5 has separate CALLs to access that peripheral specifically. For example, the serial EEPROM internal to the PicStic5 has a CALL to read the EEPROM and a different CALL to write to the EEPROM. The different CALLs and their function are listed below in Section 7.1.

Each peripheral has a set of predefined variables that the function CALL uses to complete its task. The PicBasic Pro Compiler defines these variables when the code is compiled. If the user code attempts to define these variables an error will occur when the code is compiled. A list of these variables and their function is listed below in Section 7.2.

If the supplied routines are not desired they can be omitted by compiling the user code for the PIC16F8786 processor instead of the PicStic 5 at the command prompt. Care must be taken when attempting this, as certain registers are predefined when compiling as a PicStic 5. For more information on this please contact Micromint Inc.

7.1 CALL Definitions

AD01 – Read Differential Input (CH0 = +, CH1 = -)
AD10 – Read Differential Input (CH0 = -, CH1 = +)
AIN5 – Read Channel 0 of the LTC1298 (Single Ended)
AIN6 – Read Channel 1 of the LTC1298 (Single Ended)
DACOUT – Sets and updates the both channels of the DAC
EEREAD – Reads from specified address of EEPROM
EEWRITE – Writes to specified address of EEPROM

7.2 Variable Definitions

PS5ADC – Two-byte variable that stores the result of the ADC calls
PS5EEDAT – Two byte variable that holds the data being written during the EEWRITE call and the data being read during the EEREAD call
PS5EEADD – Two-byte variable that holds the address of the memory location being written too in the EEREAD and EEWRITE calls
PS5DAC – Two byte variable that holds the 12-bit value to be written to the DAC during the DACOUT routine

COPYRIGHT

Micromint, Inc.
902 Waterway Place
Longwood, FL. 32750

DISCLAIMER

Devices sold by Micromint are covered by the warranty and patent indemnification provisions in its Terms of Sale only. Micromint makes no warranty, express, statutory, implied, or by description regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. Micromint makes no warranty of merchantability or fitness for any purposes. Micromint reserves the right to discontinue production and change specifications and process any time without notice. This product is intended for use in normal commercial applications. Applications requiring extended temperature and unusual environmental requirements, or applications requiring high reliability applications, such as military, medical life support or life-sustaining equipment, are specifically NOT recommend without processing by Micromint for such application.

Occasionally in this manual we refer to other manufacturers' products. Such reference does not constitute an endorsement of these products, but are included for the purpose of illustration or clarification. We do not intend such technical information and interface data to supersede information provided by individual manufacturers.

Conditions of Sale and Product Warranty

Micromint, Inc. and the Buyer agree to the following terms and conditions of Sale and Purchase:

1. Micromint, Inc. extends the following warranty: a factory manufactured circuit board or assembly carries with it a one-year warranty covering parts and labor. Any unit, which is found to have a defect in materials or workmanship, will, at the discretion of Micromint, Inc., be repaired or replaced.

2. Products distributed, but not manufactured by Micromint, Inc. carry the original manufacturers warranty, usually 90 days. Such products include, but are not limited to: power supplies, sensors, I/O modules, MOSARTs, LCD displays, battery-backed RAM modules, and disk drives.

3. A minimum inspection fee must be prepaid for the repair of units that are no longer under warranty. Call Micromint, Inc. for a current list of fees.

4. Micromint, Inc. will not be responsible for the repair or replacement of any unit damaged by user modification, negligence, abuse and mishandling, or improper installation.

5. Micromint, Inc. is not responsible to the Buyer for any loss or claim of special or consequential damages.

6. All units returned for repair must first receive prior authorization from Micromint, Inc.. A return authorization number can be obtained by phone, letter, or email. Please retain a record of this number, since most subsequent correspondence will refer to this authorization. Under no circumstances should any product be returned to Micromint, Inc. without such authorization, and Micromint, Inc. assumes no responsibility for returns unaccompanied by an authorization number. All returns must be shipped prepaid and should be insured. Micromint, Inc. is not responsible for losses or damage during shipment. Repaired units will be returned postage- and insurance-paid.

7. Micromint, Inc. reserves the right to alter any feature or specification at anytime. This right extends to fees, charges, and any other conditions or warranties contained herein.