ProtoCessor

FieldServer Technologies

A Sierra Monitor Company

1991 Tarob Court

Milpitas, CA 95035 USA

Toll Free: 800-317-8319

Telephone: 408-964-4433

Fax: 408-964-4425

Email: sales@ProtoCessor.com

ProtoCessor Design Guide

APPLICABILITY & EFFECTIVITY

Explains the Functioning and Set-up of the following ProtoCessor products:

FFP-485 ProtoCessor (FPC-ED2)
ASP-485 ProtoCessor (FPC-AD2)
FFP-ETH ProtoCessor (FPC-F03)
FFP-LON ProtoCessor (FPC-ED4)

The instructions are effective for all systems manufactured after December 2012

Kernel Version:	6.09
Document Revision:	0

TABLE OF CONTENTS

1 PROTOCESSOR FAMILY OVERVIEW

ProtoCessor is a family of embedded and external low cost, high performance Building and Industrial Automation multi-protocol gateways. With one part number, the ProtoCessor products are preprogrammed to automatically support 1 or multiple of the same or different of the OEM products.

This guide provides an overview of the ProtoCessor family of protocol gateway solutions and the steps required to implement each of the alternatives.

By implementing the ProtoCessor solution, instant access is gained to 100+ Industrial and Building Automation protocols. The ProtoCessor solution translates from a common Serial or Ethernet protocol on the OEM's controller to the desired field (Serial, Ethernet or LonWorks protocol).

1.1 The 3 ProtoCessor product line families

1.1.1 ProtoCessor Embedded modules

- TTL to Serial, Ethernet, and LonWorks (see section 1.4)

Marketing Name	Ordering Name	Interface Connections						Certifications	
		Serial TTL	RS-485	RS-422	Ethernet	LonWorks	KNX	BACnet BTL	LonMark
FFP-485	FPC-ED2	1	1		1			Yes	
FFP-LON	FPC-ED4	1			1	1			Yes
FFP-ETH	FPC-FO3	1			1			Yes	
ASP-485	FPC-AD2	1	1						

1.1.2 ProtoCarrier Daughter cards

- These cards are equipped with 3 ports (1 Serial, 1 Ethernet, and 1 LonWorks) (See section 1.5)

ProtoCarrier		Interface Connections						Certifications	
		RS-232	RS-485	RS-422	Ethernet	LonWorks	KNX	BACnet BTL	LonMark
Data Boards	FPC-C34		2		1			Yes	
	FPC-C35		1		1	1		Yes	Yes
	FPC-C36		1	1	1			Yes	
	FPC-C37			1	1	1			Yes
	FPC-C38	1	1		1			Yes	
	FPC-C39	1			1	1			Yes
	FPC-C40		1		1		1		
	FPC-C41				1	1	1		

1.1.3 ProtoNode External Protocol Gateways

- (see section 1.6)
- ProtoNode RER - 2 RS-485 ports and 1 Ethernet port or 1 RS-232, 1 RS-485, and 1 Ethernet port
- ProtoNode LER - 1 RS-485, 1 Ethernet port and 1 LonWorks port or
  1 RS-232, 1 Ethernet port and 1 Lonworks port

ProtoNode		Interface Connections						Certifications	
		RS-232	RS-485	RS-422	Ethernet	LonWorks	KNX	BACnet BTL	LonMark
FPC-N34	RER		2		1			Yes	
FPC-N35	LER		1		1	1		Yes	Yes
FPC-N36	RER		1	1	1			Yes	
FPC-N37	LER			1	1	1			Yes
FPC-N38	RER	1	1		1			Yes	
FPC-N39	LER	1			1	1			Yes
FPC-N40	RER		1		1		1		
FPC-N41	LER				1	1	1		

1.2 BTL Mark – BACnet Testing Laboratory

The BTL mark is a symbol that indicates to a consumer that a product has passed a series of rigorous tests conducted by an independent laboratory which verifies that the product correctly implements the BACnet features claimed in the listing. The mark is a symbol of a high-quality BACnet product. For more information about the BACnet Testing Laboratory go to: http://www.BACnetinternational.net/btl/.

1.3 LonMark Certification

LonMark International is the recognized authority for certification, education, and promotion of interoperability standards for the benefit of manufacturers, integrators and end users. LonMark International has developed extensive product certification standards and tests to provide the integrator and user with confidence that products from multiple manufacturers utilizing LonMark devices work together. FieldServer Technologies has more LonMark Certified gateways than any other gateway manufacturer, including the ProtoCessor, ProtoCarrier and ProtoNode for OEM applications and the full featured, configurable gateways.

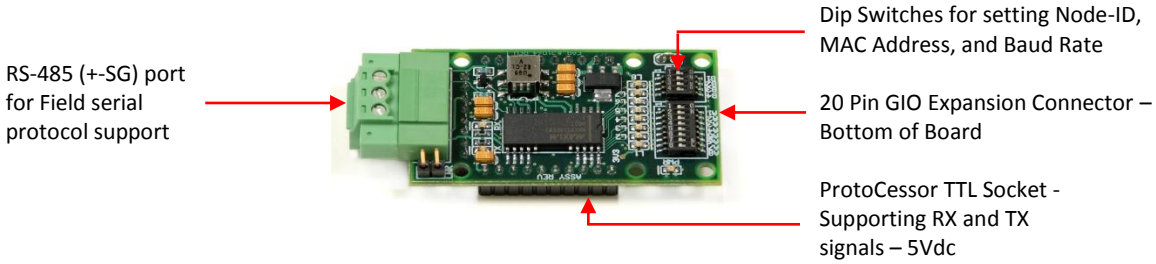
1.4 ProtoCessor Family of Modules

(Full Function ProtoCessor - FFP and Application Specific ProtoCessor - ASP): These modules are designed on the OEM's controller by implementing a ProtoCessor serial TTL socket (5VDC). See voltage ranges in section 4

Marketing Name	Ordering Name	Interface Connections						Certifications	
		Serial TTL	RS-485	RS-422	Ethernet	LonWorks	KNX	BACnet BTL	LonMark
FFP-485	FPC-ED2	1	1		1			Yes	
FFP-LON	FPC-ED4	1			1	1			Yes
FFP-ETH	FPC-FO3	1			1			Yes	
ASP-485	FPC-AD2	1	1						

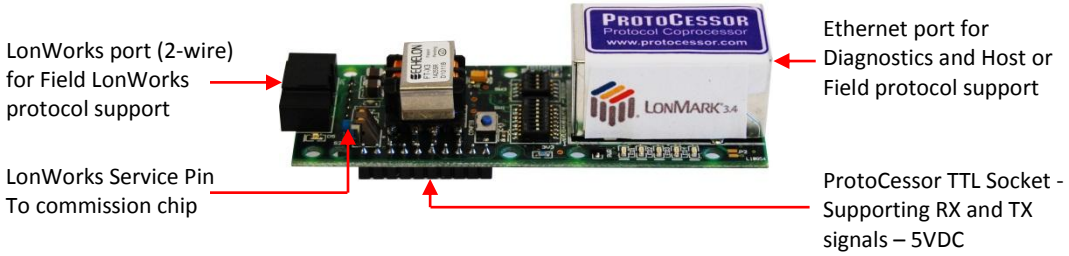
1.4.1 FFP-ETH ProtoCessor (FPC-F03)

1.4.2 ASP-485 ProtoCessor (FPC-AD2)



1.4.3 FFP-485 ProtoCessor (FPC-ED2)

1.4.4 FFP-LON ProtoCessor (FPC-ED4)



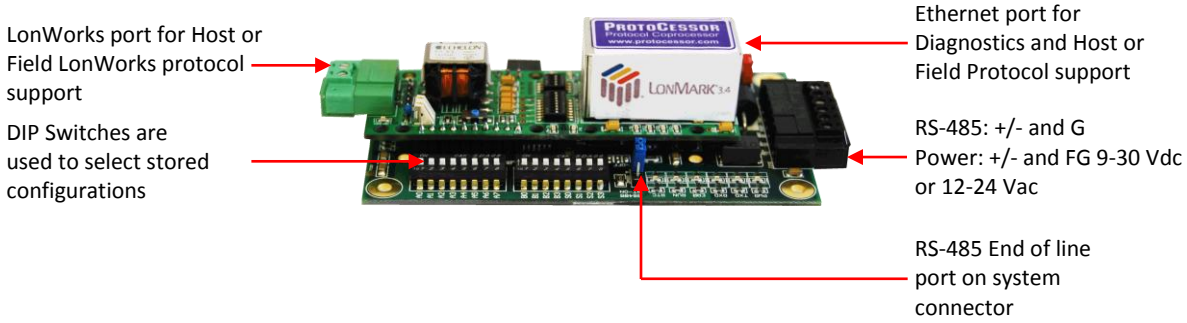
1.5 ProtoCarrier Daughter cards

These cards are designed for OEM's with boards that are not designed to accommodate a ProtoCessor TTL socket. The ProtoCarrier is able to be attached to any of the available ports (RS-232; RS-485; RS-422; Ethernet or LON, and translate to any RS-232 or RS-485 or Ethernet or LonWorks field protocol.

ProtoCarrier		Interface Connections						Certifications	
		RS-232	RS-485	RS-422	Ethernet	LonWorks	KNX	BACnet BTL	LonMark
Data Boards	FPC-C34		2		1			Yes	
	FPC-C35		1		1	1		Yes	Yes
	FPC-C36		1	1	1			Yes	
	FPC-C37			1	1	1			Yes
	FPC-C38	1	1		1			Yes	
	FPC-C39	1			1	1			Yes
	FPC-C40		1		1		1		
	FPC-C41				1	1	1		

1.5.1 ProtoCarrier FPC-C34 - 2 RS-485 and 1 Ethernet Port

1.5.2 ProtoCarrier FPC-C35 - 1 RS-485, 1 Ethernet, 1 LonWorks Port



1.5.3 ProtoCarrier FPC-C38 - 1 RS-232, 1 RS-485, 1 Ethernet Port

1.5.4 ProtoCarrier FPC-C39 - 1 RS-232, 1 Ethernet, 1 LonWorks Port

1.6 ProtoNode External Protocol Translation Gateway:

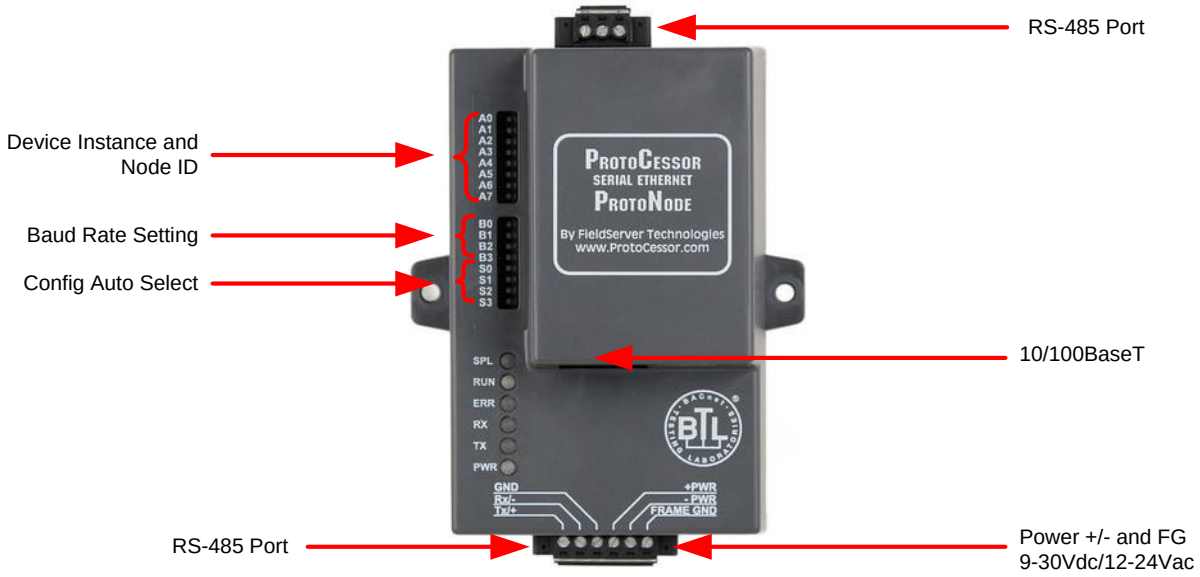
When a ProtoCessor module cannot be used and there is no place to mount the ProtoCarrier card, the alternative is to use a low cost external protocol Translation Gateway called ProtoNode. The ProtoNode family consists of 2 external solutions.

- **ProtoNode** RER - supports serial-to-serial, serial-to-Ethernet, Ethernet-to-serial, and Ethernet-to-Ethernet protocol translation.
- **ProtoNode** LER (**LonWorks**) - supports serial-to-LonWorks, Ethernet-to-LonWorks, LonWorks-to-serial, and LonWorks-to-Ethernet protocol translation.

ProtoNode		Interface Connections						Certifications	
		RS-232	RS-485	RS-422	Ethernet	LonWorks	KNX	BACnet BTL	LonMark
FPC-N34	RER		2		1			Yes	
FPC-N35	LER		1		1	1		Yes	Yes
FPC-N36	RER		1	1	1			Yes	
FPC-N37	LER			1	1	1			Yes
FPC-N38	RER	1	1		1			Yes	
FPC-N39	LER	1			1	1			Yes
FPC-N40	RER		1		1		1		
FPC-N41	LER				1	1	1		

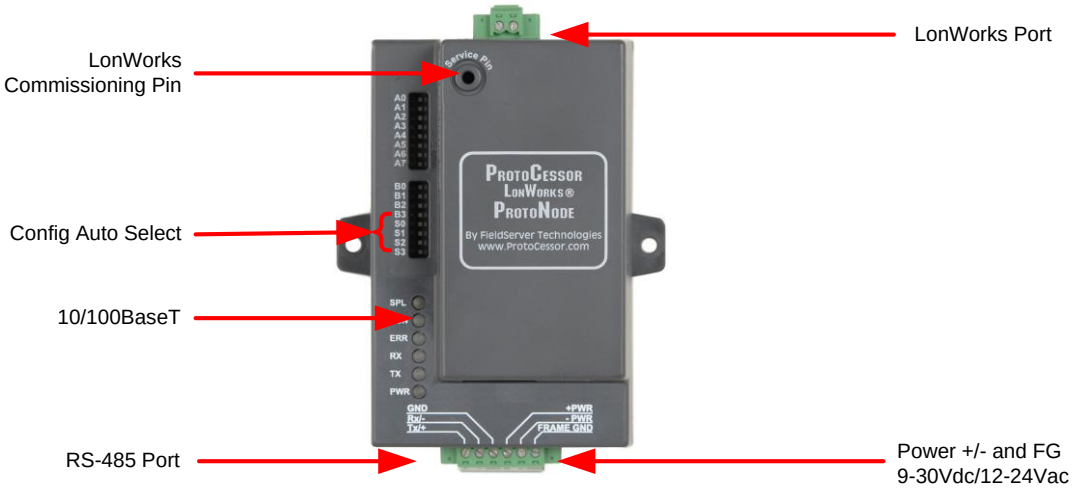
1.6.1 ProtoNode RER Components

- 1 RS-485 serial port - 3 pin screw block terminal
- 1 RJ45 10/100BaseT Ethernet port
- 1 RS-485 (FPC-N34) or RS-232 (FPC-N38) serial port, and 9-30Vdc, and 12-24Vac support on a 6 pin screw block terminal
- 3 banks of DIP switches:
  - A bank sets the Device instance, Node-ID and Mac address for MSTP
  - B bank sets Baud Rate
  - S bank is used for loading stored configurations and can store up to 250 different configurations. It can be used to enable/disable auto discovery of known devices connected to the Protonode.



1.6.2 ProtoNode-LonWorks Components

- 1 LonWorks port FFT-10
- 1 RS-485 (FPC-N35) or 1 RS-232 (FPC-N39) port
- 1 Ethernet Port
- A Banks is disabled
- S Banks is used for selected stored configurations



2 HOW PROTOCESSOR WORKS

2.1 Introduction

From a software standpoint, all three families of ProtoCessors work in the same fashion. ProtoCessor functions as an embedded gateway, enabling the OEM's equipment to rapidly utilize different protocols to interface with various Building and Industrial Automation networks. The ProtoCessor solves communication and protocol conversion problems, while enabling the OEM's to focus on their core expertise. ProtoCessor's extensive driver library provides a wide range of interoperability solutions. The way our devices work is as follows:

- We take a CSV file and we map the memory registers of the OEM's device to the various field protocols properties.
- The CSV file gets down loaded to the ProtoCessor over Ethernet and the memory registers are stored/managed in a data array inside the ProtoCessor.
- The ProtoCessor can be a master or a slave depending on what the OEM device is (master or a slave).
- We poll the OEM's device and continually update the registers in the data array. When the front end (BMS) polls us on the field protocol side, we will server up to the front the most recent data that is stored in the data array.
- This implementation allows the OEM the ability to instantly support any protocols that we support.

For the latest list of available drivers visit our website at www.ProtoCessor.com.

2.1.1 Full Function ProtoCessors (FFP)

FFPs are user configurable, have more memory, and support multiple protocols. They can support up to 1200 points mapped to the particular protocols, and all modules have an Ethernet port for remote diagnostics, and configuration.

2.1.2 Application Specific ProtoCessors (ASP)

ASP ProtoCessors have been designed specifically for OEMs with high-volume/cost sensitive products requiring efficient but affordable protocol support. The ASP has been designed for "Plug and Play" installation – no software is required. This is to ensure ease of installation and support by the OEM and their customers.

ASP Supports up to 100-150 points mapped to the particular field protocol. The ASP is programmed at the factory with a static mapping configuration which cannot be changed in the field. Several different static mappings are supported via the DIP switch user defined functions. The two banks of DIP switches enable the users to quickly configure the serial protocol settings without the need for any 3rd party software. Settings available via the DIP switches include:

- MAC address
- Baud rate (including auto-baud setting for BACnet MSTP)
- Node ID

Four special user defined functions can be selected via the DIP switches. These functions could be protocol or device related. For example, the same ASP ProtoCessor can be used on four different chiller models. The DIP switches can be used to select the specific profile used on a specific model of chiller.

ASP's have an optional 20-pin Expansion I/O Interface that includes:

- Twelve GPIO pins that can support any combination of 12 Digital I/O or Analog Inputs.
- Eight power pins (4 ground and four 3.3V pins) that can be used to power an external device up to 500 mA at 3.3V (e.g. LED's).
- To access the 20 pins, the 2O socket can be laid directly on OEM hardware or can be connected with the use of a ribbon cable (needs to be purchased separately). Refer to Appendix C for Connectors and Cables.

2.2 Application

Today's buildings and plants are integrated, intelligent facilities requiring multiple mechanical and electrical systems to be controlled from a central automation system. This central automation system is unable to decipher data from devices operating on a different protocol and therefore is unable to control these devices. The ProtoCessor provides the solution. Through its powerful protocol conversion capability the ProtoCessor allows system designers and managers to connect unique instrumentation and sensor devices onto common protocol systems. The ProtoCessor product functions as a bridge between the OEM's equipment and one or more Clients (see Figure 2.2.1).

The ProtoCessor and the OEM's device need to speak a common Host/Socket protocol. The information is gathered by this common Host/Socket side protocol of the ProtoCessor. ProtoCessor can attach the OEM's equipment via a TTL level Serial Port UART, RS-232, RS-485, Ethernet or LonWorks port.

The Socket Node Descriptor contains information about the OEM's equipment such as baud rate. The data from the OEM's equipment is stored on the ProtoCessor in a data array. The exact location as well as the format of the information is determined by the Map Descriptors. The ProtoCessor can contain any number of Data Arrays, but each Data Array can only store data in one format. The Server Map Descriptors describe how this information is able to be accessed by the Client nodes. On the Server side of the ProtoCessor, virtual nodes are created to convert the information stored in the data arrays to the format required by the Client Node. In the case of the FFPs, ProtoCarriers, and ProtoNodes; the configuration is edited in a text file which gets downloaded to the ProtoCessor over Ethernet. In the case of the ASP ProtoCessor, the configuration is hard coded at the factory.

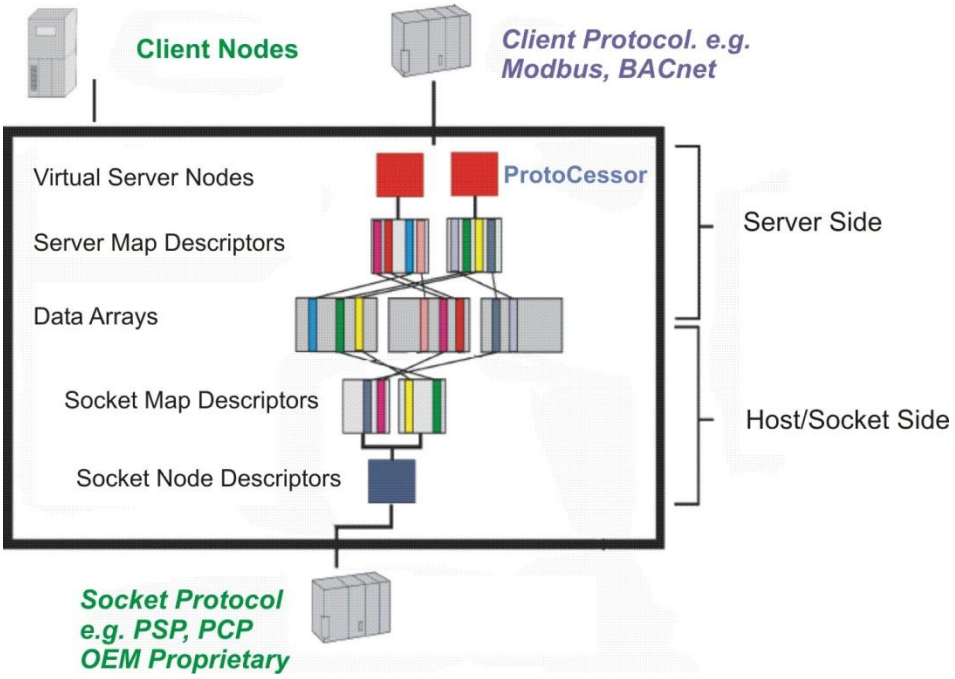


Figure 2.2.1 Theory of Operation

2.3 Configuration File for FFPs, ProtoCarriers, and ProtoNodes:

The driver configuration file (CONFIG.CSV) is in comma-delimited format which can be edited using spreadsheet programs or any text editor.

Every FFP ProtoCessor, ProtoCarrier, and ProtoNode all has an Ethernet port. The port is used for remote configuration, diagnostics, and Ethernet protocol translation.

ProtoCessor	FPC-ED2	Support 1200 points mapped out to field protocol properties
	FPC-FO3	
ProtoCarrier	FPC-C34	
	FPC-C38	
ProtoNode	FPC-N34	
	FPC-N38	

ProtoCessor	FPC-AD2	Support 100 to 150 points mapped out to field protocol properties
	FPC-ED4	Support 1000 points mapped to 1000 SNVTs
ProtoCarrier	FPC-C35	
	FPC-C39	
ProtoNode	FPC-N35	
	FPC-N39	

The CONFIG.CSV file is loaded into these devices through the Ethernet port. It can be retrieved using the FieldServer GUI (Graphic User interface) via Ethernet. Refer to the ProtoCessor webpage http://www.protocessor.com/tech_support/utilites.htm for more information. Contact FieldServer Technical Services for assistance in mapping the configuration file to a particular application.

FieldServer GUI's most significant features:

- GUI allows yout to set IP address to field protocol
- Generate fix for LonWorks network
- Transfer files (CSV configuration, firmware, etc) to and from the ProtoCessor.
- Monitor a working ProtoCessor's internal data, and parameters. Most importantly, it allows FieldServer GUI to monitor Socket communications. These are the communications to and from the ProtoCessor and the Host CPU. It displays the TX and RX packet communications, as well as the total number of bad packets.
- Change or update ProtoCessor internal data parameters.
- Delete files on a ProtoCessor.
- Restart a ProtoCessor.
- Create Serial Data Captures for diagnostics.
- View error messages.

2.4 Multiple Controllers support on a ProtoCessor

ProtoCessor sells preprogrammed/tested protocol gateways to OEMs which support all of their Building and Industrial Automation needs. Our approach insures that when an OEM sends a ProtoCessor solution to the field with their product, they know it will work every time out of the box because the configurations have all been pre-tested/validated for all their products.

The ProtoCessor protocol gateway product line has advanced functionality that makes it easy for manufacturers to configure, install, and support product in the field. ProtoCessor makes it easy to add interoperability to all the OEM's product lines to meet the demands of their customers. One part number can provide a solution that will support one or multiple of the same or different controllers to the various different field protocols. This advanced functionality means that the OEM or the integrator does not need to build or load any custom or standard configuration files to meet the OEM's different product lines that are installed in the field.

ProtoCessor's OEM testing process:

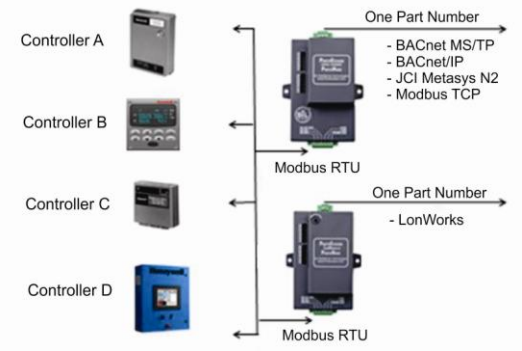
- The OEM provides the register list for all the controllers that they want BMS protocol support for.
- FieldServer programs all the OEMs different controllers for all the required protocols.
- FieldServer creates a specific part number for the OEM which will contain all the configurations that are developed for the OEM.
- When the OEM receives the first test sample they will receive all the configurations/Profiles that we developed for each of their different product lines.
- We schedule a 60 minute meeting via the phone to walk the OEM through the one time Startup/validation of the ProtoCessor device (ProtoCessor, ProtoCarrier or ProtoNode). The configurations must be validated before the ProtoCessor can be sent out to the field.
- FieldServer will provide a 90 day fully functional evaluation copy of Chipkin Automation's CAS BACnet Explorer. This program will allow the OEM to test their product on BACnet on a PC in their facility.
- FieldServer will create a user manual that the OEM can provide to their customers that explains how to install their products on the various protocols. The OEM can use the manual as it is or incorporate it into their own style.

Once the validation is complete, FieldServer then takes the validated Configurations/Profiles for each of the OEM's controllers and completes and freezes the programming for the final ProtoCessor production configuration. ProtoCessor offers three approaches of a final configured ProtoCessor OEM gateway. The best approach will depend on the OEM's requirements (multiple families of controllers and multiple protocols). See the 3 approaches below.

Configuration Auto-Select: (http://www.protocessor.com/products/Configuration-Auto-Selector.php)

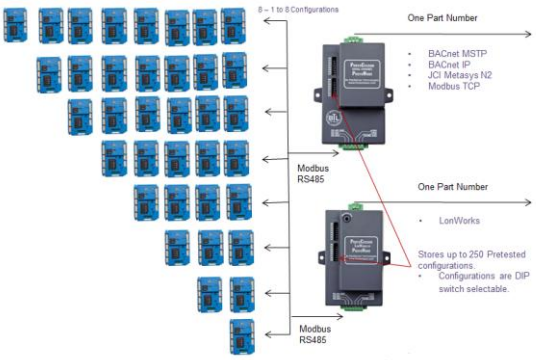
Configuration Auto-Select means that all pretested configurations are already loaded onto the ProtoCessor gateway and are selectable via DIP switches. Various combinations of configurations are developed and loaded onto the ProtoCessor. Various possibilities include:

1. A common device protocol interfacing to multiple protocols – for instance a single device with Modbus RTU communication can have access to various protocols such as BACnet MS/TP, BACnet/IP, Metasys N2 for JCI, Modbus TCP or LonWorks



2. Multiple devices interface to a common protocol – the manufacturer has multiple products that need to communicate to BACnet/IP, thus the ProtoCessor has preloaded multiple configurations from Devices A, B, C or D to BACnet/IP. A dip switch selects the correct configuration

3. Multiple of the same types of devices interface to multiple protocols – a combination of the two in which the manufacturer has multiple of the same devices and they need to interface to a variety of protocols. Again, a DIP switches selects the correct device and protocol combination and loads it.

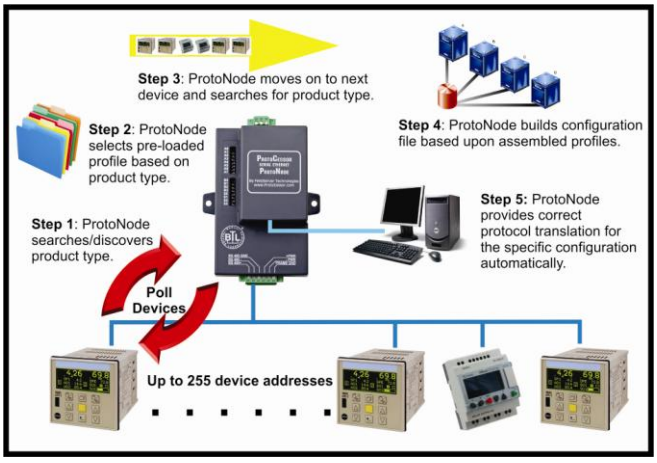


Advanced Auto-Discovery (http://www.protocessor.com/products/Advanced-Auto-Discovery.php)

ProtoCessor Advanced Auto-Discovery is for applications that require 1 or multiple of the same or devices connect to one ProtoNode needing to support multiple Field Protocols without having to build any special configurations. The Configurations files are built automatically in the field.

The ProtoCarrier/ProtoNode will search and discover any recognizable Profiles that are stored inside the ProtoCarrier/ProtoNode. We can store up to 250 device profiles inside the ProtoCarrier/ProtoNode. Each profile needs to have a unique register that we can use to identify the device or we can use Modbus 17 (Slave ID request) to discover any know profiles if the device supports Modbus function 17.

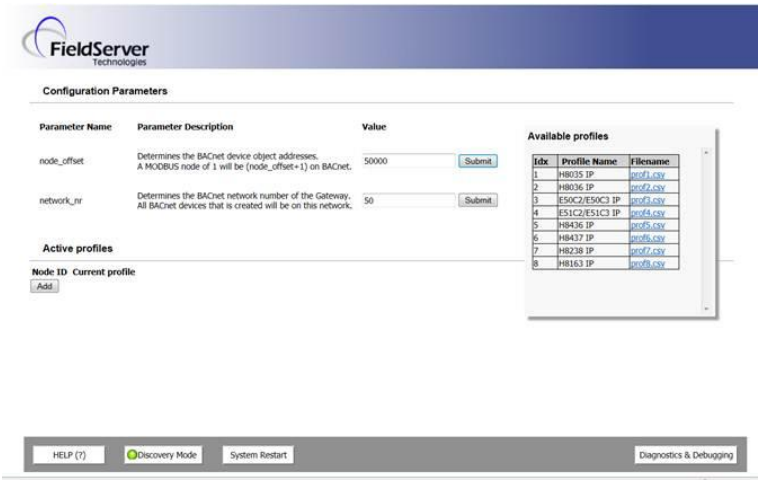
- On Power up, the ProtoCarrier/ProtoNode will poll device addresses 1 to 255. Each Profile will take its turn to read its unique register to see if it can be identified. If a Profile recognizes a device, the ProtoCarrier/ProtoNode loads that particular profile in memory and moves to the next device address.
- Polling will continue until the point limitation has been reached (i.e. 1200 Modbus points) or if all device addresses have been polled (up to 255). Once all devices are discovered, the ProtoNode will automatically build and load the configuration file.



- Profiles are preloaded into the ProtoCarrier/ProtoNode for each of the OEM's products needing to be discovered.
- Once it completes the entire polling cycle, it will build the configuration file for all the devices discovered and automatically load the file. Setting the S3 DIP switch to off saves the configuration that was just built and the Product is installed in the desired Field Protocol.

Profile Selector/Web Configuration (http://12.49.212.118)

For devices that do not have a unique identifying register, the ProtoNode can be set-up using the Profile Selector/Web Configurator to select the specific device profile that are already stored inside the ProtoNode. This solution can support one or multiple of the same or different controllers connected to the ProtoNode needing support for all the required field protocols. Via the web you can also add device profiles to the "available profile" list.



1. The user simply goes to the IP address for the specific ProtoNode and it will open the Configuration Parameters screen.
2. Select "Add" and choose from the list of available device profiles, add in the Node ID and save
3. The Configuration file is automatically generated from the profiles selected.

3 PROTOCESSOR SUPPORTED HOST & FIELD PROTOCOL COMMUNICATIONS

OEMs need to select a common host side protocol that the ProtoCessor can understand. ProtoCessor supports a wide range of legacy host protocols (like Modbus), but for OEM devices that do not have host protocol, 2 alternatives are available:

- Implement our PSP ASCII protocol. (ProtoCessor Simple Protocol). It takes about 1 day to implement. Refer to Appendix A for protocol spec.
- If the OEM has proprietary host protocol, FieldServer can write the driver on the ProtoCessor Host/Socket platform for an NRE fee.

The following table lists the currently supported OEM's Host/Socket Protocols. The list of supported protocols is constantly increasing, and it is advisable to contact ProtoCessor or refer to the website for a more updated list.[1]

3.1 Table Showing Sample List of Supported Protocols

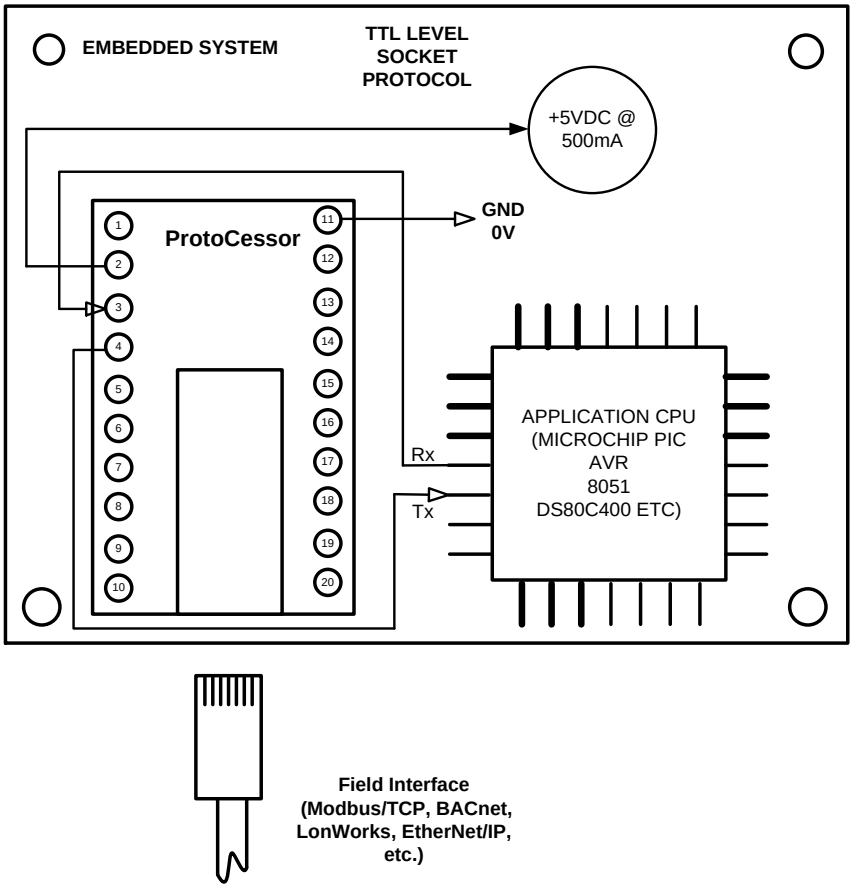
OEM's Host Serial or Ethernet Protocols	Serial Field Protocols	Ethernet Field Protocols	FieldBus Protocols
Modbus RTU	BACnet MSTP	Modbus TCP	LonWorks
Modbus ASCII	BACnet PTP	Allen Bradley DF1	KNX (Q4 2012)
ProtoCessors PSP Driver	Modbus RTU	DNP 3.0	
BACnet MSTP	Modbus ASCII	BACnet/IP	
Metasys N2 Open	BACnet MSTP	BACnet Ethernet	
XML	Metasys N2 Open	EtherNet/IP	
AB DF1	AB DF1	Allen Bradley CSP	
Modbus TCP	DNP3 Serial	DNP3 Ethernet	
BACnet/IP		GE-SRTP	
LonWorks		GE-EGD	
EtherNet/IP		Omron	
Allen Bradley CSP		SNMP	
DNP 3.0 Serial or Ethernet		XML	
GE-SRTP			
GE-EGD			
SNMP			
OEM's Custom Serial Driver			
KNX - Q4 2012 - ProtoNode/ProtoCarrier			

[1] Visit www.ProtoCessor.com for the complete list of supported protocols.

4 IMPLEMENTING THE PROTOCESSOR MODULES - HARDWARE

4.1 The ProtoCessor Socket:

The OEM needs to implement a ProtoCessor socket on the board consisting of u-shaped 2 x 10, and 1 x 8 pin headers (reserved pins) which include the TX/RX signals power supply. This socket will accommodate any current and future ProtoCessor. The socket is populated only when the need for the protocol exists.



4.1.1 Pin Assignment for ProtoCessor Socket for 2x10 Way Strips

PIN #	Function	Direction	DTE Label	PIC32 PIN	Comments
1	Frame Ground (FG)				Not DC Ground (0V)
2	+5V				
3	TX	From ProtoCessor	TxD (out)	RF 05	
4	RX	To ProtoCessor	RxD (in)	RF 04	
5	CTS	To ProtoCessor	CTS (in)	RB 08	
6	RTS	From ProtoCessor	RTS (out)	RB 14	
7	DIO1		DSR (in)	RG 06	Alternative I/O: SCK
8	DIO2		DTR (out)	RB 06	Alternative: PGC
9	DIO3		DCD (in)	RB 07	Alternative: PGD
10	Reserved				Alternative: nRESET
11	0V				Circuit Ground
12	SCL			RG 02	I2C Clock line
13	SDA			RG 03	I2C Data Line
14	DIO4		RI (in)	RG 08	Alternative I/O: SDO
15	DIO6			RF 01	Active High I/O GPIO
16	DIO7			RF 00	Active High I/O GPIO
17	DIO5			RG 07	Alternative I/O: SDI
18	Reserved				Field Programming
19	Reserved				
20	Reserved				

Serial Peripheral Interface (SPI) is supported with signals: SDI, SDO, SCK

4.1.2 ProtoCessor Pin Voltage Levels

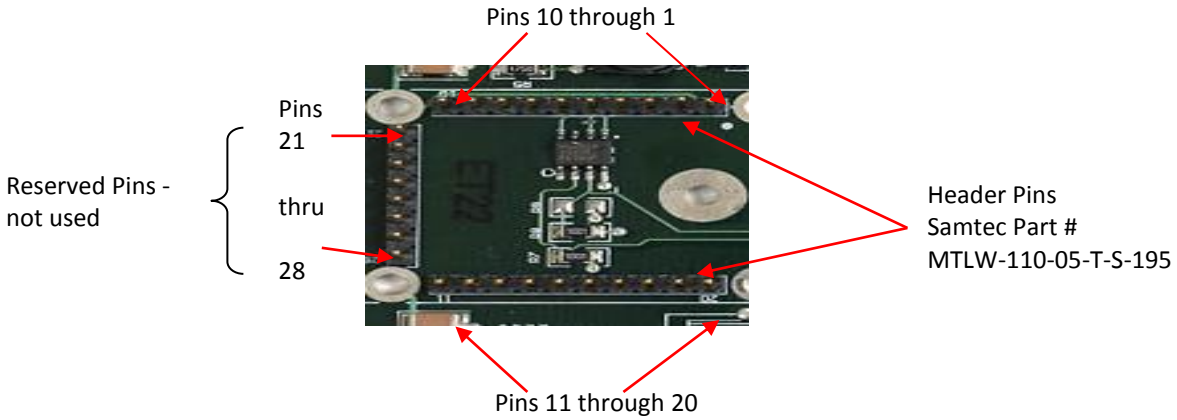
Description	Pin #	Min	Max	Unit
Input (Rx) High Voltage	4	2	5.3	V
Input (Rx) Low Voltage	4	0	0.66	V
Output (Tx) High Voltage	3	2.4	3.45	V
Output (Tx) Low Voltage	3	0	0.4	V

4.1.3 ProtoCessor Pin Headers

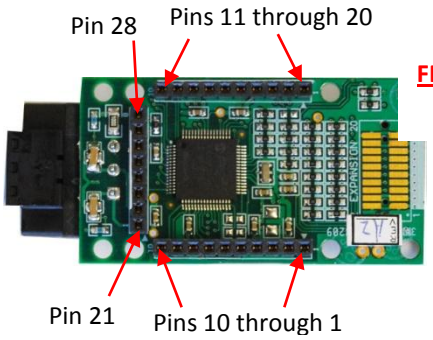
ProtoCessor recommends use of the following SAMTEC Pin Headers on host board:

Part Numbers:	TLW-1xx-x-S or MTLW-1xx-x-S
Manufacturer:	SAMTEC
Link to Data Sheets:	http://www.samtec.com/documents/webfiles/pdf/TLW_TH.PDF
	http://www.samtec.com/documents/webfiles/pdf/MTLW_TH.PDF

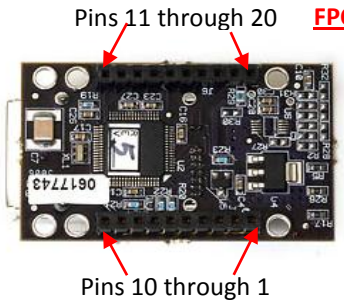
4.2 ProtoCessor Socket Pin Locations



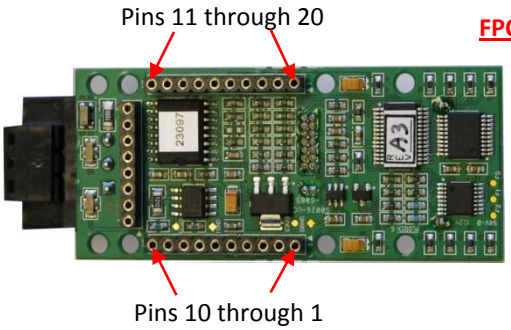
4.3 Location of Pins on the ProtoCessor:



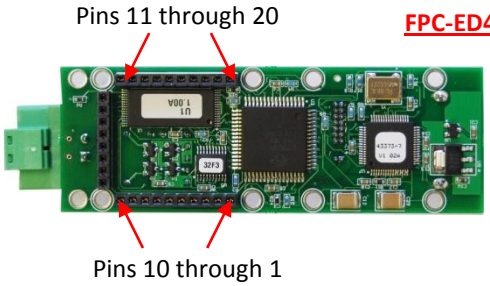
FPC-ED2 (ASP-485)



FPC-FO3 (FFP-ETH)



FPC-AD2 (FFP-485)



FPC-ED4 (FFP-LON)

5 PROTOCESSOR POWER REQUIREMENTS[2]

Power Requirement for ProtoCessor ASP at 3.3V through 5 VDC					
Standalone ProtoCessor			Combined with ProtoCarrier 485 V2		
ProtoCessor Type Mk 2	3.3VDC	5 VDC	12VDC/VAC	24VDC/VAC	30VDC
FPC-AD2 (Typical)	100mA	120mA	80mA	40mA	40mA
FPC-AD2 (Maximum)	120mA	140mA	90mA	50mA	50mA
Note: These values are 'nominal' and a safety margin should be added to the power supply of the host system. A safety margin of 25% is recommended.					

Power Requirement for ProtoCessor FFP at 5 VDC				
Standalone ProtoCessor		Combined with ProtoCarrier 485 V2		
ProtoCessor Type Mk 2	5 VDC	12VDC/VAC	24VDC/VAC	30VDC
FFP-ED2 (Typical)	480mA	180mA	100mA	100mA
FFP-ED2 (Maximum)	490mA	280mA	150mA	120mA
FPC-ED4 (Typical)	480mA	210mA	100mA	90mA
FPC-ED4 (Maximum)	490mA	250mA	130mA	100mA
Note: These values are 'nominal' and a safety margin should be added to the power supply of the host system. A safety margin of 25% is recommended.				

Power Requirement for ProtoCarrier at 9V through 30 VDC or 12-24 VAC			
	Current Draw Type		
ProtoCarrier Family	12VDC/VAC	24VDC/VAC	30VDC
FPC – C34, C36, C38 (Typical)	170mA	100mA	80mA
FPC – C34, C36, C38 (Maximum)	240mA	140mA	100mA
FPC – C35, C37, C39 (Typical)	210mA	100mA	90mA
FPC – C35, C37, C39 (Maximum)	250mA	130mA	100mA
Note: These values are 'nominal' and a safety margin should be added to the power supply of the host system. A safety margin of 25% is recommended.			

Power Requirement for ProtoNode at 9V through 30 VDC or 12-24 VAC			
	Current Draw Type		
ProtoNode Family	12VDC/VAC	24VDC/VAC	30VDC
FPC – N34, N36, N38 (Typical)	170mA	100mA	80mA
FPC – N34, N36, N38 (Maximum)	240mA	140mA	100mA
FPC – N35, N37, N39 (Typical)	210mA	100mA	90mA
FPC – N35, N37, N39 (Maximum)	250mA	130mA	100mA
Note: These values are 'nominal' and a safety margin should be added to the power supply of the host system. A safety margin of 25% is recommended.			

[2] Visit www.ProtoCessor.com for the latest information.

Appendix A. PROTOCESSOR SIMPLE PROTOCOL (PSP) SPECIFICATION

Appendix A.1. Purpose of the Protocol

The ProtoCessor Simple Protocol (PSP) is recommended in instances where the ProtoCessor is the Server and the OEM CPU is the Client.

- Implemented on the "Host" or "Application" CPU on the OEM circuit board (Microcontroller).
- Defined as an ASCII protocol in order to facilitate debugging, analysis, and implementation.
- Allows data to be transferred to and from the ProtoCessor
- Allows the Host Microcontroller to configure aspects of the ProtoCessor e.g. IP address, Node ID, Baud rate
- Data that is written to the ProtoCessor is available to other devices on the ProtoCessor Ethernet connection. Data written to the ProtoCessor by other devices over the Ethernet protocol is available to be read by the Host CPU.

Appendix A.2. Protocol Description

Appendix A.2.1. Quick start – demonstrates the simplicity of the application:

Assuming a ProtoCessor is installed with a Factory Default configuration where there are no configuration commands necessary; there is no need to configure any aspects of the ProtoCessor unless the application justifies the additional complexity.

The first point of data from the ProtoCessor is the "Read Data". The transaction between the Application Microcontroller and the ProtoCessor would appear as follows:

From Host CPU/MCU	:RD-NA:data,OF:0<cr>
Response from ProtoCessor:	:27.3<cr>

The 27th value in the ProtoCessor is the "Write Data". The transaction would appear as follows:

From Host CPU/MCU	:WD-NA:data,OF:26,VA:23.9<cr>
Response from ProtoCessor:	:OK<cr>

The FMT command in the poll from the Host CPU to the ProtoCessor can be used to specify the format of the data traveling to and from the ProtoCessor. If no FMT is specified, then the ProtoCessor returns a floating point value.

From Host CPU/MCU	:RD-NA:data,OF:0,FMT:H<cr>
Response from ProtoCessor:	:0x89CA<cr>

Appendix A.2.2. Serial Parameters

The PSP has the following default serial parameter settings:

BAUD	38400
Data Bits	8
Parity	N
Stop Bits	1

Appendix A.2.3. Message Structure

Every PSP Protocol Packet has the same basic structure:

Start	Payload	Terminator
:	{see Appendix A.2.5)	<CR>

Appendix A.2.4. Binary Packed Messages

Binary Packed messages were considered but the implementation of these was rejected for the following reasons.

Byte Order

Floating Point Format

Floating Point Order

As an alternative to this it will be possible to use block reads for faster data transfer. Another option is to use the RCOV functionality implemented in this protocol.

Appendix A.2.5. Payload Contents

Payload Packets can generally be defined as: Command – parameter 1, parameter 2,

Three basic types of commands exist:: Transferring Data, Setting Parameters and Issuing Control Commands.

Transferring Data Commands

		Required Parameter			Optional Parameter
Description	Command	Parameter 1	Parameter 2	Parameter 3	Parameter 4
Read Data from a Data Array in the ProtoCessor	RD	Data Array Name	Data Array Offset	-	FMT (default to decimal unsigned integer)
Write data to a Data Array in the ProtoCessor	WD	Data Array Name	Data Array Offset	value	FMT (default to decimal unsigned integer)
Block Read (Future)	RB	Data Array Name	Data Array Offset	Data Length	
Change Of Value Read	RCOV	-	-	-	
Change of Value Read and Ack	RCOVA	-	-	-	
Simple COV ACK	A or ACK	-	-	-	

Parameters

Description	Long form	Abbreviation	Example	Possible Values
Data Array Name	Name:{parameter}	NA:{parameter}	:RD-NA:data,OF:0<cr>	
Data Array Offset	Offset:{0-10000}	OF:{0-10000}	:RD-NA:data,OF:12<cr>	
Value	Value:{parameter}	VA:{parameter}	:WD-NA:data,OF:2,VA:2.9<cr>	
Format	FMT:{parameter}	FMT:H	:WD-NA:data,VA:1000,FMT:D :WD-NA:data,VA:0xab3,FMT:H :WD-NA:data,VA:1000,FMT:U :WD-NA:data,VA:10.1,FMT:F	D – Signed Decimal Int H – Hexadecimal Integer U - Unsigned Integer F – Floating Point

Setting Communication Parameters

Description	Long form	Abbreviation	Example	Possible Values
Field Protocol Node ID	Node ID:{01-255}	ND:{01-255}	:WD-ND:125678<cr>	Any 32-bit Value
ProtoCessor Mac Address	Mac ID:{01-255}	MD:{01-255}	:WD-MD:11<cr>	Any 8-bit Value
Field Protocol Baud Rate	Baud:{300-38400}	BD:{300-38400}	:WD-BD:9600<cr>	9600, 38400, 57600

Control commands

Description	Long form	Abbr	Example	Comments
Enable ProtoCessor Node	Node_Enable	NE	:NE<cr>	Node will Enable automatically 30 seconds after ProtoCessor startup.
Disable ProtoCessor Node	Node_Disable	ND	:ND<cr>	The Node can be disabled by default at startup using a PSP Connection parameter.

Status commands

Description	Long form	Abbreviation	Example	Possible Results
Get ProtoCessor Status	Node_Status	NS	:NS<cr>	:ND<cr> – Node Disabled :NE<cr> – Node Enabled :TR<cr> – Trouble/Panic :CE<cr> – Config Error :SD<cr> – Startup Delay

Appendix A.2.6. Writing to Output Data Objects

Output objects can be written to using the PSP protocol only if the Node has been disabled. If a Host system wants to initialize (or default) the ProtoCessor's output objects, the following steps commands must be executed:

Host systems checks to see when the ProtoCessor started by doing a NS command.

```
:NS<cr>
```

The host system can expect one of the following conditions in response to a NS command.

No response This will happen while the ProtoCessor is still starting up.

"ND<cr>" The Node is disabled.

Host system writes default values to the Output objects.

```
WD-NA:DA_REG,OF:0,VA:23.9<cr>
```

Host system enables the ProtoCessor by issuing the NE command.

```
:NE<cr>
```

Host system checks the ProtoCessor status.

```
:NS<cr>
```

Appendix A.3. Error Conditions

Appendix A.3.1. Format of the Error Response

Errors are reported by the ProtoCessor as follows:

```
From Host CPU                :RD-NA:data,OF:0<cr>
Response from ProtoCessor:   :ERR0001<cr>
```

This means error condition 1 has occurred. See the Error Table for details.

Appendix A.3.2. Error Response Table

Error Number	Error Description
0001	Data Array does not exist
0002	Data Array Offset does not exist
0003	Illegal Format code for FMT - keyword
0004	Corrupted Message
0005	Configuration update (Baud, Node_Id etc) failed.

Appendix A.4. Using Change of Value Reads

If it is necessary to acquire updated data quickly from a system that has a large number of points then RCOV is the most effective method to do this. The PSP Server will return the first data block that has been updated.

```
From Host CPU                :RCOV<cr>
Response from ProtoCessor    :RD-NA:data,OF:x,VA:y<cr>
```

Where: data -> is the data array name.

x -> offset within that data array.

y -> value of the data.

The format of the data returned will be the basic data type of that point on the server.

In order for the PSP server to know that this RCOV arrived at the client correctly the client needs to send an acknowledgement.

```
From Host CPU                :A<cr> or :ACK<cr>
Response from ProtoCessor    :No response from ProtoCessor, waits for next message.
```

This then clears the COV flag for that particular piece of data.

The host CPU can also ACK the data with an RCOVA:

From Host CPU :RCOVA<cr>
Response from ProtoCessor :RD-NA:data,OF:x,VA:y<cr>

As well as clearing the (now read) COV flag this also allows the server to respond with the next piece of updated data that it finds. If the server cannot find any further updated data, it will respond with an OK which signals to the client that all the updated data has been read. The client can then resume normal RCOV polls until it, once again, receives updated data.

From Host CPU :RCOVA<cr>
Response from ProtoCessor :OK<cr>

Appendix B. PIN ASSIGNMENT FOR PROTOCESSOR[3]

Appendix B.1. Pin Assignment for ProtoCessor Socket for 2x10 Way Strips

PIN #	Function	Direction	DTE Label	PIC32 PIN	Comments
1	Frame Ground (FG)				Not DC Ground (0V)
2	+5V				
3	TX	From ProtoCessor	TxD (out)	RF 05	
4	RX	To ProtoCessor	RxD (in)	RF 04	
5	CTS	To ProtoCessor	CTS (in)	RB 08	
6	RTS	From ProtoCessor	RTS (out)	RB 14	
7	DIO1		DSR (in)	RG 06	Alternative I/O: SCK
8	DIO2		DTR (out)	RB 06	Alternative: PGC
9	DIO3		DCD (in)	RB 07	Alternative: PGD
10	Reserved				Alternative: nRESET
11	0V				Circuit Ground
12	SCL			RG 02	I2C Clock line
13	SDA			RG 03	I2C Data Line
14	DIO4		RI (in)	RG 08	Alternative I/O: SDO
15	DIO6			RF 01	Active High I/O GPIO
16	DIO7			RF 00	Active High I/O GPIO
17	DIO5			RG 07	Alternative I/O: SDI
18	Reserved				Field Programming
19	Reserved				
20	Reserved				
Serial Peripheral Interface (SPI) is supported with signals: SDI, SDO, SCK					

Appendix B.2. Pin Assignment for 1x8 Way Socket Strip on all ProtoCessors

PIN #	Function	Direction	DTE Label	Comments
21	RS-485 + (ISO)	I/O		Passthrough Connection from Terminal Block
22	RS-485 - (ISO)	I/O		Passthrough Connection from Terminal Block
23	RS-485 GND (ISO)	GND		Passthrough Connection from Terminal Block
24	RS-232-TX	OUT		Reserved (Not used)
25	RS-232-RX	INT		Reserved (Not used)
26	RS-232-RTS	IN		Reserved (Not used)
27	RS-232-CTS	OUT		Reserved (Not used)
28	SPARE	I/O		Reserved (Not used)

[3] For latest revision, check www.ProtoCessor.com

Appendix B.3. Pin Assignment - ASP FPC-AD4 and FPC-AD5 Expansion Connector

PIN #	PIN NAME	Direction	PIC32 PIN	Type 1	Type 2	Type 3	Type 4
1	DGND	GROUND	-	-	-	-	-
2	DGND	GROUND	-	-	-	-	-
3	3.3V	POWER	-	-	-	-	-
4	3.3V	POWER	-	-	-	-	-
5	EXT_IO11	I/O	RB 13	GPIO	AI 13	TDI	-
6	EXT_IO12	I/O	RB 15	GPIO	AI 15	CN 12	OCFB
7	EXT_IO09	I/O	RB 11	GPIO	AI 11	TDO	-
8	EXT_IO10	I/O	RB 12	GPIO	AI 12	TCK	
9	EXT_IO07	I/O	RB 09	GPIO	AI 09	-	C2 OUT
10	EXT_IO08	I/O	RB 10	GPIO	AI 10	TMS	C VREF
11	EXT_IO05	I/O	RB 04	GPIO	AI 04	CN 6	C1 IN-
12	EXT_IO06	I/O	RB 05	GPIO	AI 05	CN 7	C1 IN+
13	EXT_IO03	I/O	RB 02	GPIO	AI 02	CN 4	C2 IN-
14	EXT_IO04	I/O	RB 03	GPIO	AI 03	CN 5	C2 IN+
15	EXT_IO01	I/O	RB 00	GPIO	AI 00	CN 2	VREF +
16	EXT_IO02	I/O	RB 01	GPIO	AI 01	CN 3	VREF -
17	3.3V	POWER	-	-	-	-	-
18	3.3V	POWER	-	-	-	-	-
19	DGND	GROUND	-	-	-	-	-
20	DGND	GROUND	-	-	-	-	-

GPIO = General Purpose I/O (5V tolerance TTL levels)

AI = Analogue Input (8-10 Bit Discrete Values)

CN = Change Notification (Interrupt Signals to PIC32)

C1 / C2 / CVREF = Comparator Inputs / Outputs / Voltage Level (Optional)

VREF + / VREF- = A/D Converter High / Low Reference Voltages (Optional)

Appendix C. RECOMMENDED CONNECTORS, CABLES AND MECHANICS

Appendix C.1. ProtoCessor Pin Headers

ProtoCessor recommends use of the following SAMTEC Pin Headers on host board:

Part Numbers:	TLW-1xx-x-S or MTLW-1xx-x-S
Manufacturer:	SAMTEC
Link to Data Sheets:	http://www.samtec.com/documents/webfiles/pdf/TLW_TH.PDF http://www.samtec.com/documents/webfiles/pdf/MTLW_TH.PDF

Appendix C.2. Expansion Connector For ASP - FPC-AD4 and FPC-AD5

ProtoCessor recommends use of the following connector on the host board:

Header Part Number:	FTS-110-01-F-DV
Manufacturer:	SAMTEC
Link to Data Sheet:	http://www.samtec.com/documents/webfiles/pdf/FTS.PDF

Appendix C.3. Expansion Cable For ASP[4] – FPC-AD4 and FPC-AD5

ProtoCessor recommends use of the following cables on the expansion connector:

Cable Part Number:	FFSD-10-S-10.00-01-N (Single Ended Ribbon)
Cable Part Number:	FFSD-10-D-10.00-01-N (Double Ended Ribbon)
Manufacturer:	SAMTEC
Link to Data Sheet:	http://www.samtec.com/documents/webfiles/pdf/FFSD.PDF

ProtoCessor recommends use of the following cables on the expansion connector:

Cable Part Number:	SFSD-10-28-H-10.00-S
Cable Part Number:	TFSD-10-28-H-10.00-D-NUS
Manufacturer:	SAMTEC
Link to Data Sheet:	http://www.samtec.com/documents/webfiles/pdf/sfsd.pdf http://www.samtec.com/documents/webfiles/pdf/tfsd.pdf

Appendix C.4. Mechanics

This spacer will provide the required 0.312” spacing as per Appendix F.

Part Number:	27DMSP00250
Manufacturer:	Micro Plastics, Inc
Link to Website:	https://secure.microplastics.com/default.aspx

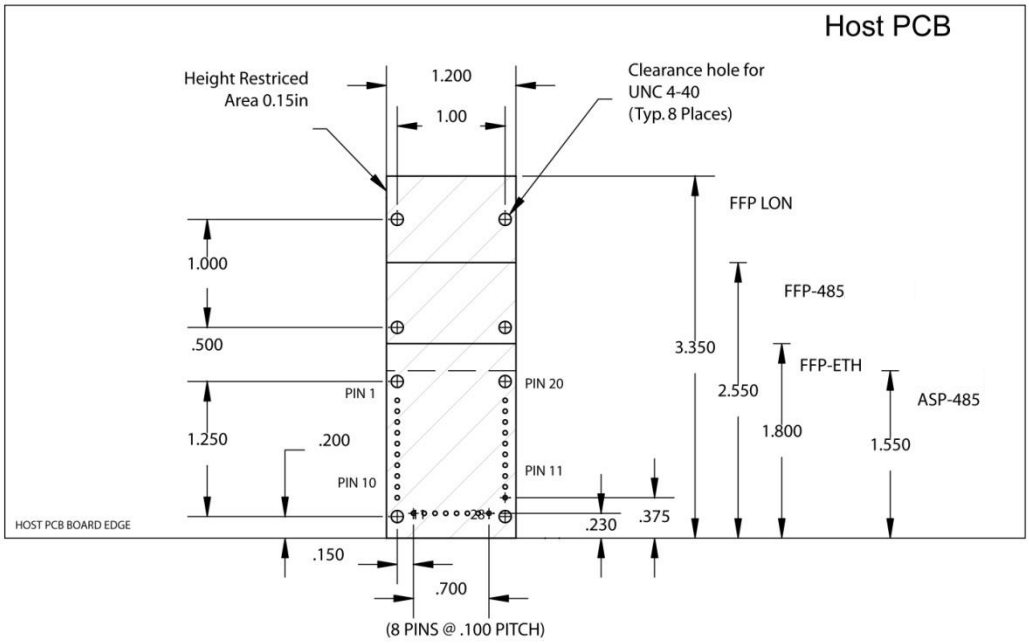
[4] Expansion Cable for ASP - Available on request (Cable only works in combination with right angle expansion connector on ASP board)

Appendix D. SUPPORTED ASP PROTOCESSOR HARDWARE VERSIONS:

Part Number	Voltage Span	Host Port	Field Port	Expansion Connector	GPIO
FPC-AD2	4.5V - 7,5VDC	Un-Isolated TTL	RS-485 Optically Isolated	No	No
FPC-AD3	3.0V - 3.6VDC	Un-Isolated TTL	RS-485 Optically Isolated	No	No
FPC-AD4	4.5V - 7,5VDC	Un-Isolated TTL	RS-485 Optically Isolated	Yes - 20 Pin	12 GPIO – Digital I/O of Analog inputs
FPC-AD5	3.0V - 3.6VDC	Un-Isolated TTL	RS-485 Optically Isolated	Yes - 20 Pin	12 GPIO – Digital I/O of Analog inputs

Appendix E. PROTOCESSOR SOCKET PCB FOOTPRINT

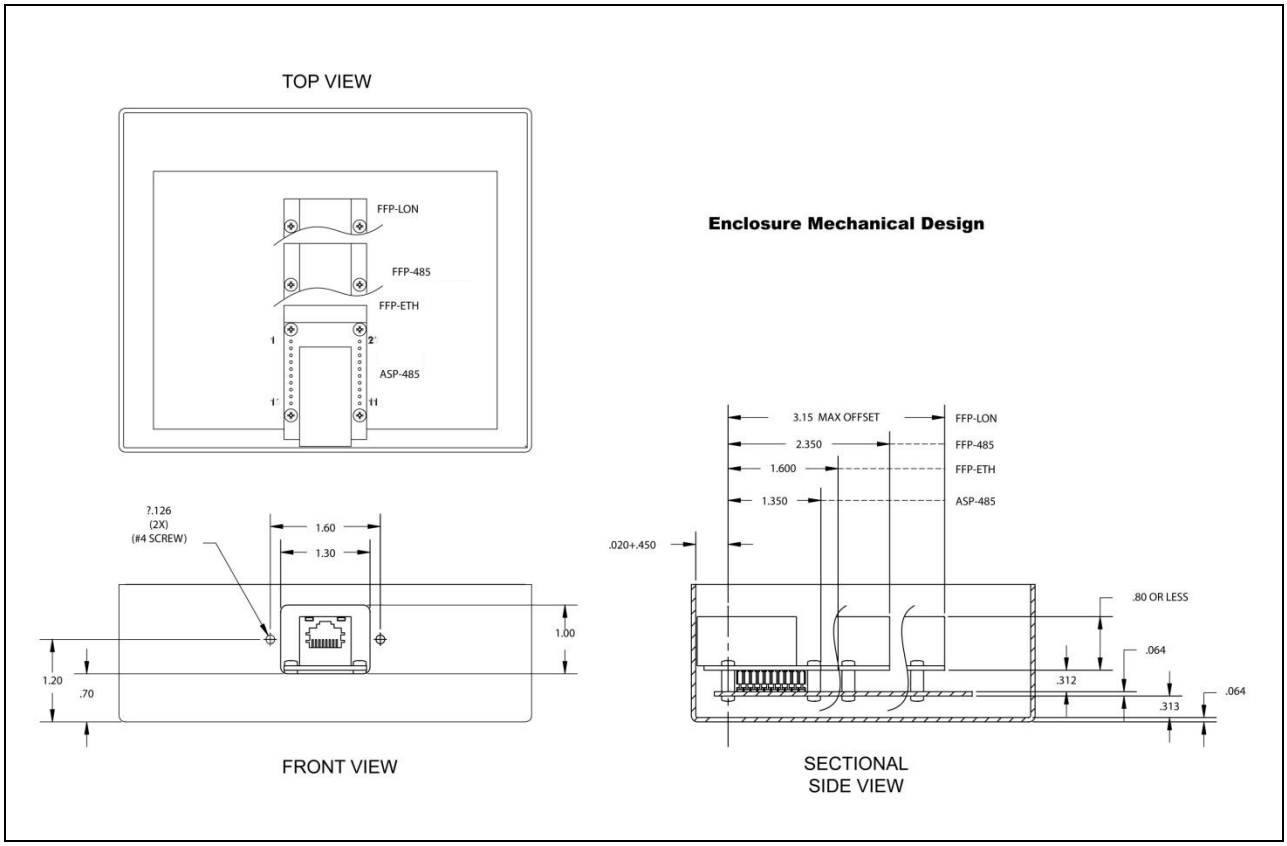
PCB FOOTPRINT AND KEEP-OUT AREA



NOTES:

THIS IS NOT THE SIZE OF THE PROTOCESSOR MODULE, BUT THE SIZE OF THE PCB FOOTPRINT.
FOR SOCKET: SAMTEC PN: TLW-110-05-G-S

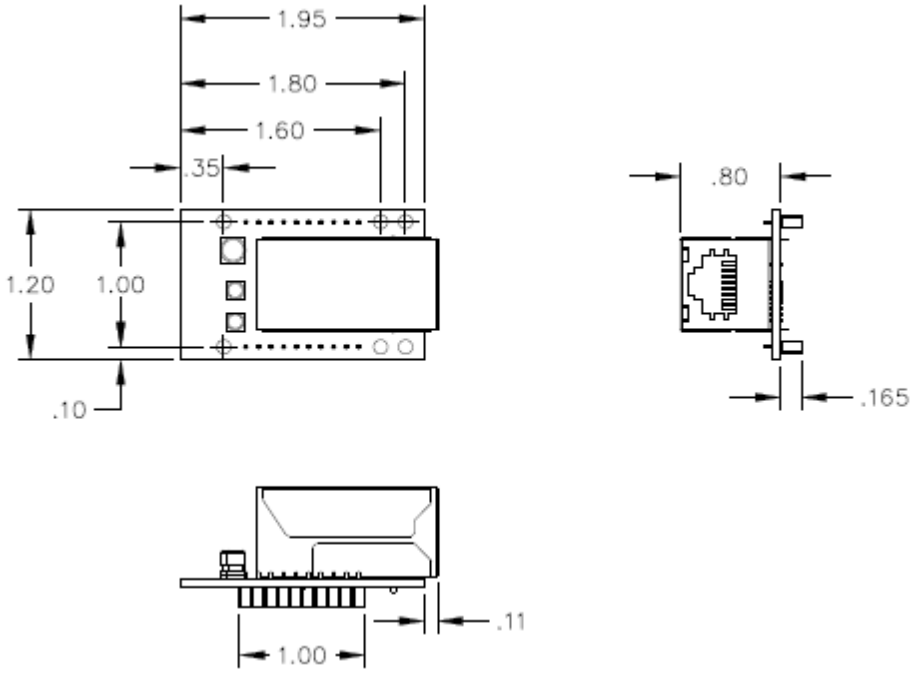
Appendix F. ENCLOSURE – MECHANICAL DESIGN[5]



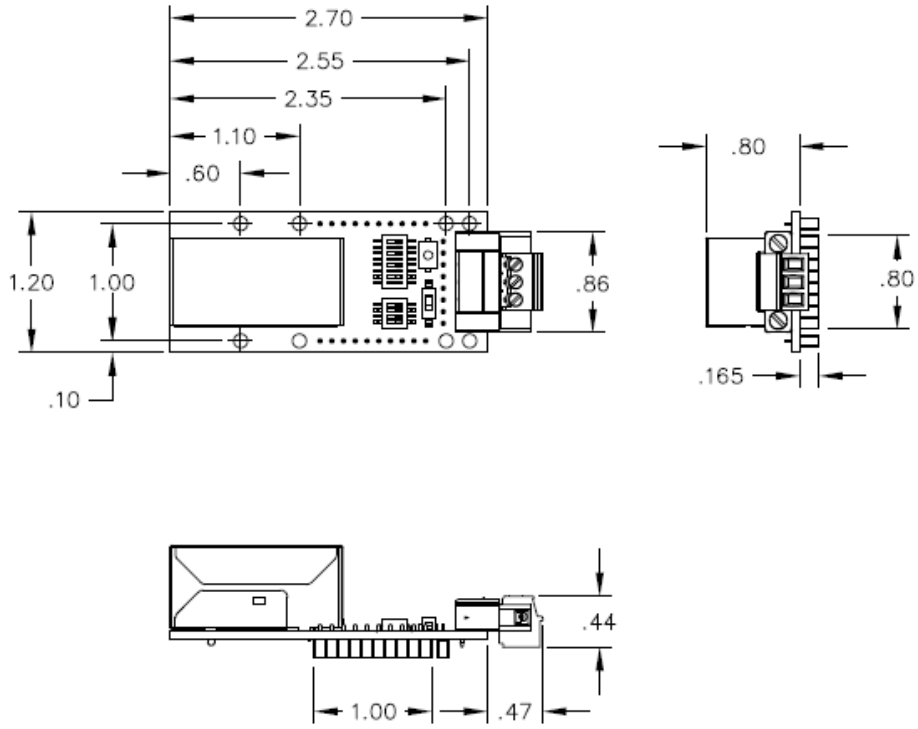
[5] the height of 0.312" is only applicable when using the spacer as per Appendix C.4

Appendix G. MECHANICAL DRAWINGS – PROTOCESSOR AND PROTOCARRIER

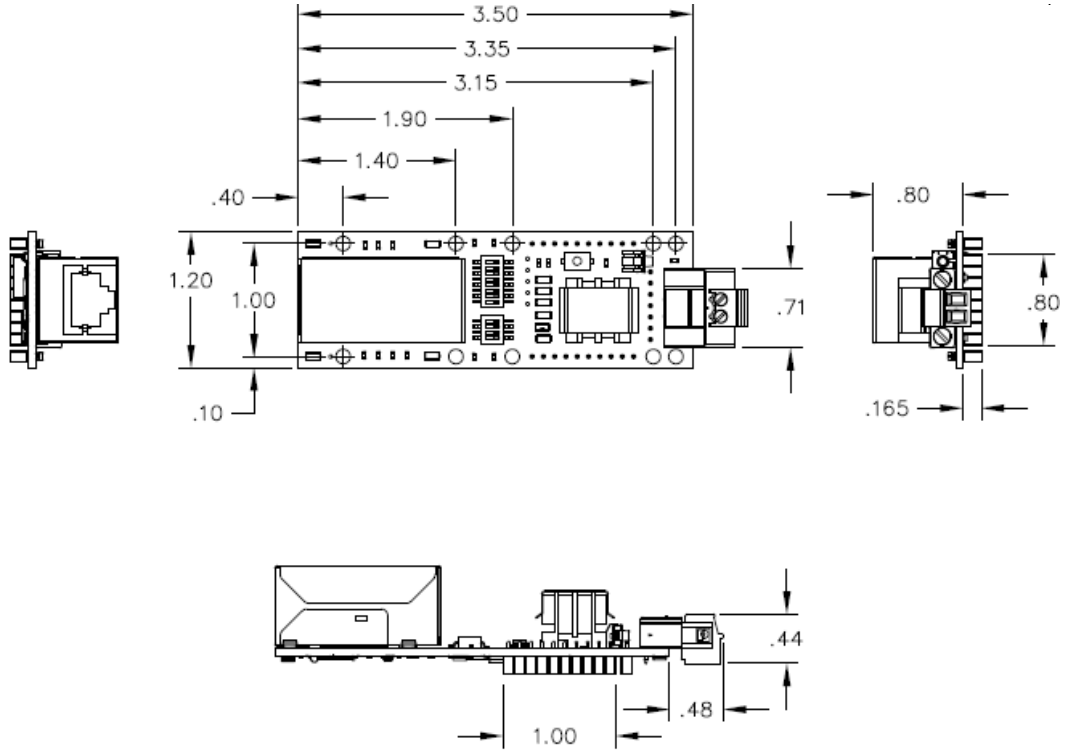
Appendix G.1. Mechanical Dimension Drawing FPC-FO3



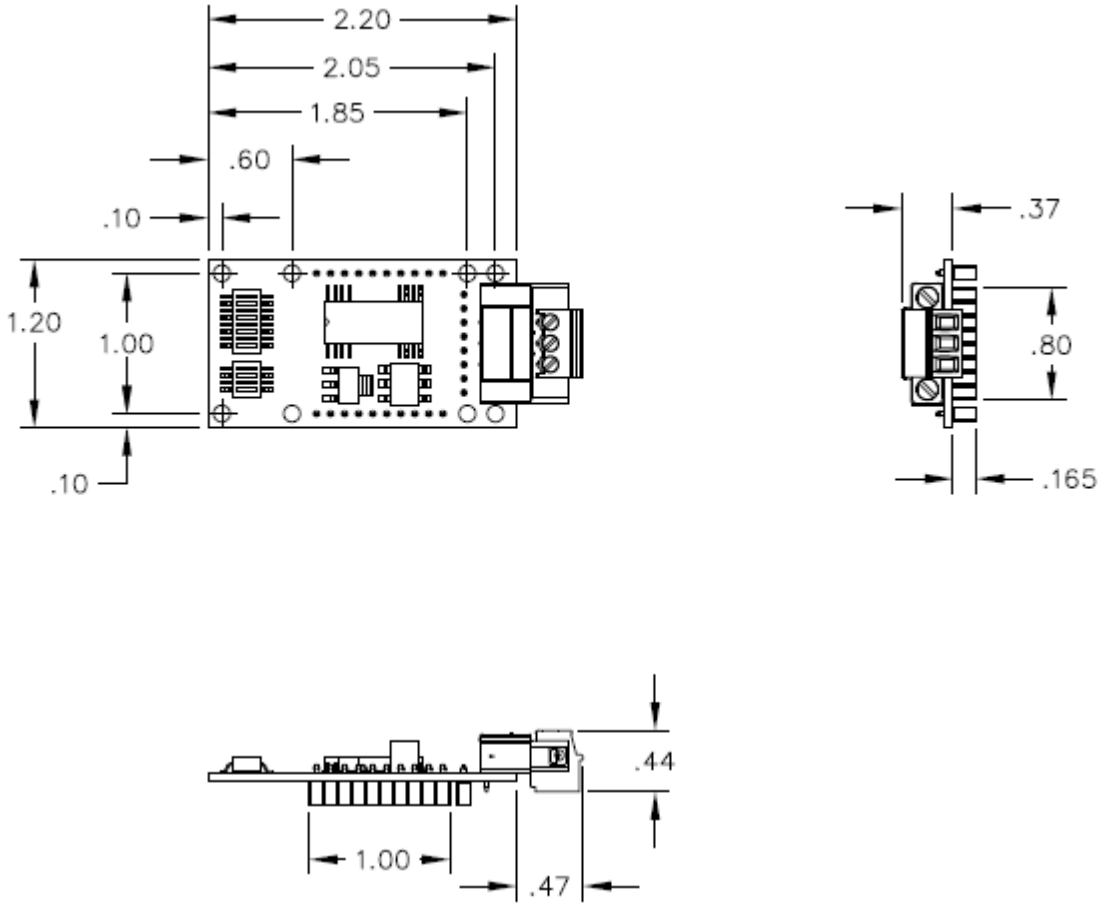
Appendix G.2. Mechanical Dimension Drawing FPC-ED2



Appendix G.3. Mechanical Dimension Drawing FPC-ED4

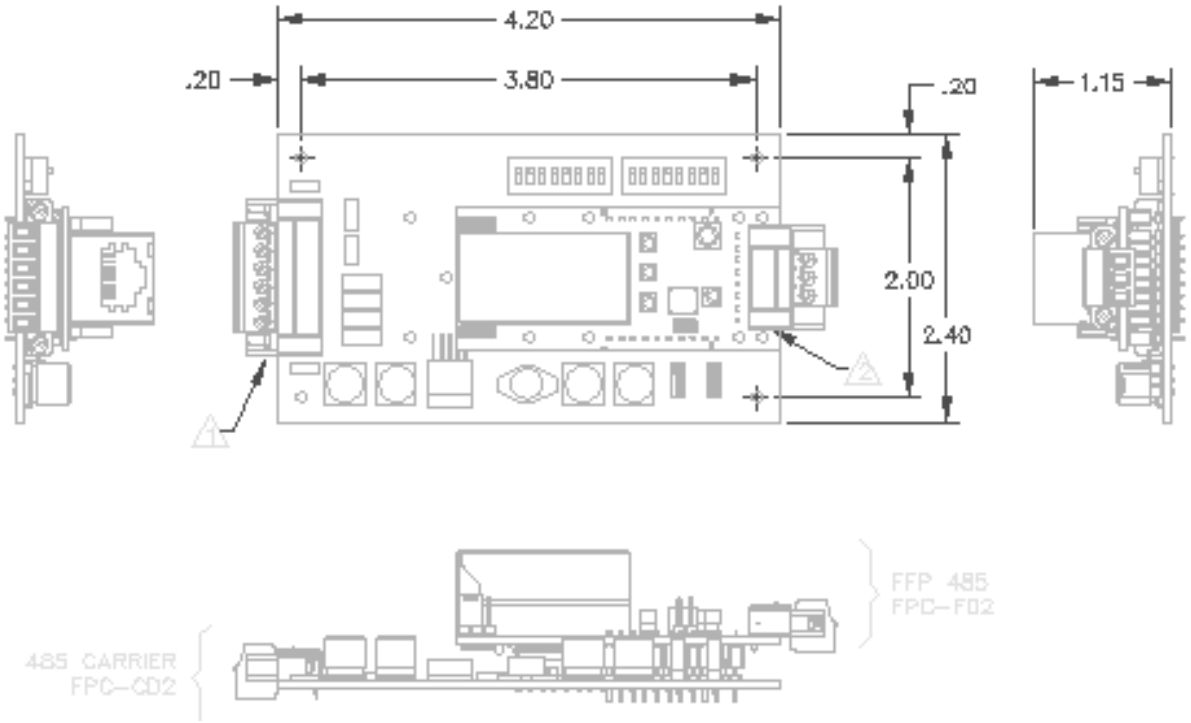


Appendix G.4. Mechanical Dimension Drawing FPC-AD2



Appendix H. PROTOCESSOR MOUNTING ON PROTOCARRIER – DIMENSION DRAWINGS

Appendix H.1. Mounting the ProtoCarrier – FPC-C34, C38 & C40



Appendix H.2. ProtoCarrier Mounting FPC-C35, C39 & C41