

# Acuity<sup>®</sup>

## AccuProfile™ 820 Laser Scanners User's Manual



Manual p/n LLL008201 – Rev. 3.4  
For use with Acuity products  
March 15, 2017

Acuity  
A product line of Schmitt Industries, Inc.  
2765 NW Nicolai St.  
Portland, OR 97210  
[www.acuitylaser.com](http://www.acuitylaser.com)

# Limited Use License Agreement

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THE PACKAGE CONTAINING THE COMPUTER SOFTWARE AND HARDWARE LICENSED HEREUNDER. CONNECTING POWER TO THE MICROPROCESSOR CONTROL UNIT INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD PROMPTLY RETURN THE UNIT WITH POWER SEAL INTACT TO THE PERSON FROM WHOM IT WAS PURCHASED WITHIN FIFTEEN DAYS FROM DATE OF PURCHASE AND YOUR MONEY WILL BE REFUNDED BY THAT PERSON. IF THE PERSON FROM WHOM YOU PURCHASED THIS PRODUCT FAILS TO REFUND YOUR MONEY, CONTACT SCHMITT INDUSTRIES INCORPORATED IMMEDIATELY AT THE ADDRESS SET OUT BELOW.

Schmitt Industries Incorporated provides the hardware and computer software program contained in the microprocessor control unit, and licenses the use of the product to you. You assume responsibility for the selection of the product suited to achieve your intended results, and for the installation, use and results obtained. Upon initial usage of the product your purchase price shall be considered a nonrefundable license fee unless prior written waivers are obtained from Schmitt Industries Incorporated.

## LICENSE

- a. You are granted a personal, nontransferable and non-exclusive license to use the hardware and software in this Agreement. Title and ownership of the hardware and software and documentation remain in Schmitt Industries, Incorporated;
- b. the hardware and software may be used by you only on a single installation;
- c. you and your employees and agents are required to protect the confidentiality of the hardware and software. You may not distribute, disclose, or otherwise make the hardware and software or documentation available to any third party;
- d. you may not copy or reproduce the hardware and software or documentation for any purpose;
- e. you may not assign or transfer the hardware and software or this license to any other person without the express prior written consent of Schmitt Industries Incorporated;
- f. you acknowledge that you are receiving only a LIMITED LICENSE TO USE the hardware and software and related documentation and that Schmitt Industries Incorporated retains title to the hardware and software and documentation. You acknowledge that Schmitt Industries Incorporated has a valuable proprietary interest in the hardware and software and documentation.

YOU MAY NOT USE, COPY, MODIFY, OR TRANSFER THE HARDWARE AND SOFTWARE, IN WHOLE OR IN ANY PART, WITHOUT THE PRIOR WRITTEN CONSENT OF SCHMITT INDUSTRIES, INCORPORATED.

IF YOU TRANSFER POSSESSION OF ANY PORTION OF THE HARDWARE OR SOFTWARE TO ANOTHER PARTY, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

## TERM

The license is effective until terminated. You may terminate it at any other time by returning all hardware and software together with all copies of associated documentation. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to return the hardware and software together with all copies of associated documentation. In the event of termination the obligation of confidentiality shall survive.

## 12 MONTH LIMITED WARRANTY

EXCEPT AS STATED BELOW IN THIS SECTION THIS PRODUCT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Schmitt Industries Incorporated does not warrant that the functions contained in the product will meet your requirements or that the operation of the product will be uninterrupted or error free.

Schmitt Industries Incorporated does warrant as the only warranty provided to you, that the product which is furnished to you, will be free from defects in materials and workmanship under normal use for a period of twelve (12) months from the date of delivery to you as evidenced by a copy of your warrant receipt.

## LIMITATIONS OF REMEDIES

Schmitt Industries Incorporated's entire liability and your exclusive remedy shall be:

1. the replacement of any hardware and software not meeting Schmitt Industries' "Limited Warranty" and which is returned to Schmitt Industries Incorporated or an authorized Schmitt Industries dealer with a copy of your purchase receipt, or
2. if Schmitt Industries Incorporated or the dealer is unable within ninety

(90) days to deliver a replacement product which is free of defects in material or workmanship, you may terminate this Agreement by returning the product and your money will be refunded to you by the dealer from whom you purchased the product.

IN NO EVENT WILL SCHMITT INDUSTRIES INCORPORATED BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH PRODUCTS EVEN IF SCHMITT INDUSTRIES INCORPORATED OR AN AUTHORIZED DEALER HAD BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

SOME AREAS DO NOT ALLOW THE LIMITATIONS OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

## GENERAL

You may not sublicense, assign or transfer the license or the hardware, software, and documentation except as expressly provided in this Agreement. Any attempt otherwise to sublicense, assign or transfer any of the rights, duties or obligations hereunder is void.

This Agreement will be governed by the laws of the United States and the State of Oregon, United States of America.

Should you have any questions concerning this Agreement, you may contact Schmitt Industries Incorporated by writing to:

Schmitt Industries Incorporated 2765 NW Nicolai St. Portland, Oregon 97210 USA

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU FURTHER AGREE THAT IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN YOU AND SCHMITT INDUSTRIES INCORPORATED AND ITS DEALER ("US") WHICH SUPERSEDED ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN US RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT

# Procedures for Obtaining Warranty Service

---

1. Contact your Acuity distributor or call Schmitt Industries, Inc. to obtain a return merchandise authorization (RMA) number within the applicable warranty period. Schmitt Industries will not accept any returned product without an RMA number.
2. Ship the product to Schmitt Industries, postage prepaid, together with your bill of sale or other proof of purchase, your name, address, description of the problem(s). Print the RMA number you have obtained on the outside of the package.

This device complies with:

FCC Part 15, Class A, CE Mark

CISPR22, CNS13438, EN55022, ICES-003, VCCI

Operation is subject to the following two conditions:

(1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this device in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

This manual copyright © 2013, Schmitt Industries, Inc.

# Table of Contents

---

- Procedures for Obtaining Warranty Service ..... 1
- Table of Contents ..... 2
- Table of Figures ..... 4
- 1. Introduction..... 5
  - 1.1 General Overview ..... 5
- 2. Definition of Terms..... 6
  - 2.1 Quick Start Instructions ..... 6
    - 2.1.1 Mounting and Power connections..... 6
    - 2.1.2 Ethernet connections..... 7
    - 2.1.3 Activation ..... 7
- 3. General Description ..... 8
  - 3.1 Principles of Operation ..... 8
  - 3.2 Mechanical Dimensions ..... 8
  - 3.3 Contents ..... 12
  - 3.4 Laser Safety..... 12
  - 3.5 Sensor Maintenance ..... 13
  - 3.6 Sensor Service..... 13
  - 3.7 Sensor Specifications ..... 13
- 4. Installation and Checkout..... 14
  - 4.1 Mounting ..... 14
  - 4.2 Cabling ..... 14
    - 4.2.1 Control cable ..... 14
    - 4.2.2 Ethernet cable ..... 15
  - 4.3 Connection box (Optional) ..... 15
  - 4.4 Power On..... 16
    - 4.4.1 Communications Check ..... 16
    - 4.4.2 Verifying Operation ..... 16
- 5. Ethernet Interface Operation ..... 17
  - 5.1 Changing the IP-address..... 17
- 6. Transmission Formats Sensor Data..... 18
  - 6.1 Scanner data..... 18
  - 6.2 Info-Telegram..... 19

7.	Data Formats.....	21
7.1	Profile data.....	21
7.2	Encoder Position Data.....	21
7.3	Register of Functions .....	22
7.4	Sensor temperature.....	22
7.5	Scanner Status Register.....	22
7.6	Version Evaluation Registers.....	22
7.7	Operating Hours Counter .....	22
7.8	Switch-on Counter.....	23
7.9	Input Status.....	23
7.10	Adjusting the laser output.....	23
7.11	Number of Horizontal and Vertical Pixels.....	23
7.12	FIFO-First In, First Out.....	24
7.13	Serial number .....	24
7.14	Z and X values Info-Telegram address 106 to 125.....	24
7.15	Sensor Data .....	24
7.16	Version Firmware.....	24
8.	Scanner control .....	26
8.1	Data format for Register-addresses and Data .....	26
8.2	Register of functions.....	26
9.	Description of the control commands .....	28
9.1	Exposure control .....	28
9.2	Calculation of the Exposure Time .....	28
9.3	Limiting the Range Z .....	28
9.4	Amplification.....	28
9.5	AGC (amplification control).....	29
9.6	Ambient Light Filter .....	29
9.7	Synchronization Mode .....	29
9.8	Measurement Control Trigger.....	29
10.	RS232 Interface.....	30
11.	Description of the Web-Servers.....	31
11.1	Changing the IP-address .....	31
12.	Programming using Windows DLL Calls.....	32
12.1	Importing EthernetScanner.dll.....	32
12.2	Calling EthernetScanner_GetVersion.....	33

12.3	Establishing Ethernet communication with the AP820.....	33
12.4	Calling EthernetScanner_Connect, Ethernet_Scanner_Disconnect, and EthernetScanner_GetConnectStatus.	.34
12.5	Getting AP820 Parameters by calling EthernetScanner_GetInfo.....	35
12.6	Getting Profile Data from the AP820 .....	36
12.6.1	Examples EthernetScanner_GetScanRawData.....	37
12.7	Description of other parameters returned by EthernetScanner_GetScanRawData ..	39
12.8	Writing Commands to the AP820 .....	39
12.8.1	Examples: Laser on, Triggering .....	40
12.8.2	Example: Limiting the Range Z.....	41
12.9	Description of Register Addresses accessible via EthernetScanner_WriteData .....	42

## Table of Figures

---

Figure 1	AP820-20 mechanical dimensions .....	9
Figure 2	AP820-40 mechanical dimensions .....	9
Figure 3	AP820-60 mechanical dimensions .....	10
Figure 4	AP820-80 mechanical dimensions .....	10
Figure 5	AP820-120 mechanical dimensions .....	10
Figure 6	AP820-240 mechanical dimensions .....	11
Figure 7	AP820-400 mechanical dimensions .....	12
Figure 8	AP820 laser labels .....	13
Figure 9	LED Configuration.....	16

# 1. Introduction

---

This section is a guide to getting started with the AP820 laser sensor and this manual. The AP820 scanner has a number of configurable parameters, a minimum of which must be set before the scanner can be used the first time. However, once the parameters are set, the sensor will default to that configuration until changed by the user.

The recommended order for reading the manual is:

- General Overview – Gives a brief understanding of the sensor operation.
- Operating Guidelines – Provides a few important safety tips.
- Definition of Terms – An aid for proper communication.
- Quick Start Instructions – This should provide the information necessary to connect the sensor and verify its operation with an Ethernet connection.
- General Description – Gives important laser, operation, mechanical, and mounting information.
- Installation and Checkout – Tailor the application. Use the other chapters for reference:
  - Signal and Power Interface – how to connect to the laser scanner
  - Ethernet Interface Operation – modes and formats
  - Performance Optimization – Sample Rate, Background Elimination, Exposure control

## 1.1 General Overview

---

The Acuity AccuProfile 820 Laser Line Sensor measures surface heights by projecting a beam of visible laser light that creates a line on a target surface. Reflected light from the surface is viewed from an angle by a two-dimensional CCD detector array inside the AP820 sensor. The 2D contour profile is calculated by the scanner's microprocessor from the pixel data from the diffusely reflected laser line. The height profile is transmitted through Ethernet communications to a PC computer. Real-time 3D profiling is created by synchronizing the position of the scanner with encoder inputs from conveyors, linear stages or robotic movements.

A variety of models are specified, each to allow a different measurement range and field of view. The AP820 scanner technical data sheet specifies scanner performance standards.

A variety of configuration settings can be selected via the Ethernet interface. The complete list of settings is found in the AP820 Command Set chapter and each setting is discussed in detail in a specific operation chapter. A DLL Manual also exists for programming information.

Measurement output is in the form of ASCII data packets for Ethernet models.

The scanner's detector uses a CCD-2D camera, a line laser and the electronics for the processing of the sensor's data signals.

The scanner provides a 2D scan of an object on the measurement range. The width of the scan will be indicated with an X-value, the depth of the measurement will be indicated with a Z-value.

A linearization of the scanner has been carried out in the factory and is stored in the sensor. Profile data are output in metric units - mm's. A calibration by the user is not

necessary. Therefore every scanner can be exchanged on-site, without concern for recalibration.

## 1.2 Operating Guidelines – Safety Issues

---

**Do not point the line laser at any person, particularly a person’s eyes or face.**

**Do not attempt to disassemble the scanner. Improper disassembly will destroy the optical alignment of the sensor and necessitate factory repairs.**

**Do not operate the sensor in areas where the sensor case is exposed to direct sunlight for extended periods or where the air temperature is more than 40°C (104°F) or less than 0°C 32°F).** Special order scanners include an air / water cooling capabilities for hot environments up to 400 °C.

**Avoid excessive vibration and shocks.** The scanner contains securely-mounted, but precisely-aligned optical components.

**Do not scratch the lenses on the front face of the scanner.** Keep the lenses clean with expert optical procedures. The lenses are glass with an anti-reflection coating. Avoid the use of organic cleaning solvents.

**Do not touch the lenses with bare fingers.** The oils are very difficult to remove.

Operate only with DC supply voltages up to 30 volts.

## 2. Definition of Terms

---

Scanner, Sensor – The complete 2D profile measuring device.

Target – The object of measurement. The relative distances from the scanner to the target surface is measured by the sensor.

Laser, Laser line – This light (visible laser line) is emitted from the sensor, reflected from the target, and collected by the camera lens.

<Range> – The maximum relative distance measurable by the sensor in the Z axis.

Range Beginning, Range Ending- The region over which the target can be measured. At the near end of the range the sensor measures zero. At the far end of the range the sensor measures its maximum value (its Range value).

Field of View – The lateral distance profiled by the scanner in the X axis. The Field of View varies at different Ranges.

### 2.1 Quick Start Instructions

---

This will get the AP820 scanner operating for quick measurements in its factory default configuration. For full instructions use the individual sections of this manual for mounting, power connections, data connections and configuration.

#### 2.1.1 Mounting and Power connections

---

Mount the scanner in such a way that the case is not twisted or warped. Use two screws through the threaded mounting holes on the sides of the AP820 scanner. See the technical data sheet or the section of this User’s Manual for mounting hole specifications. The laser should be aimed at a target such that the distance from the sensor face to the target can be measured, between the *Range Beginning* and *Range End*.

Attach the Power / Data cable's M12-A-coded, 8-pin female plug to the male

receptacle on the top of the scanner while the power is off.

Connect the Supply + (White) and Return (Green) wires of the sensor cable to a 10 to 30 volt DC power supply (or use the sensor's power supply if it came with one).

## 2.1.2 Ethernet connections

---

Attach the Ethernet cable's 4-pin female plug to the male receptacle on the top of the scanner.

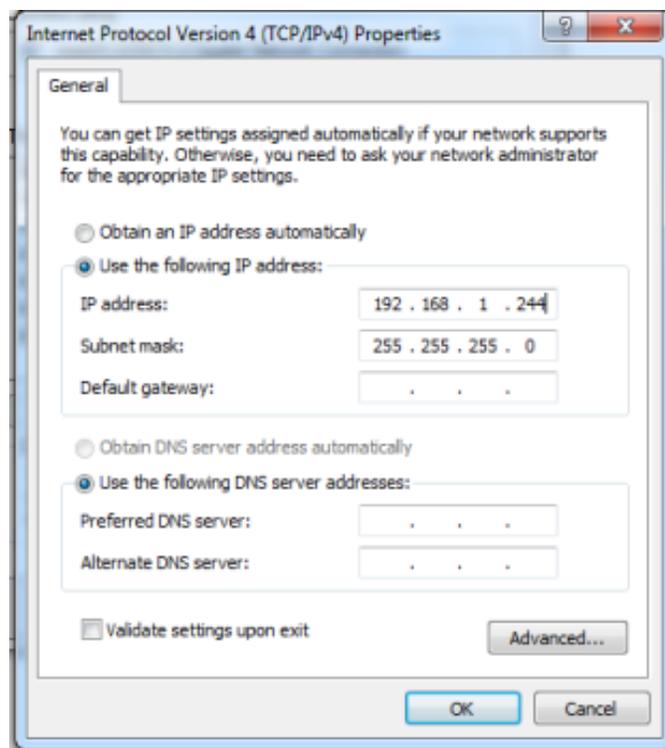
Connect the Ethernet cable's RJ45 connector to an available Ethernet port on a PC computer.

## 2.1.3 Activation

---

Plug the AP820's Ethernet plug into the Ethernet socket on the PC, and power up the laser. The AP820 has a default IP address of 192.168.1.245 - Port 1096. Subnet Mask is 255.255.255.0. For sensors older than 4/2014 please use the IP address of 192.168.123.222 with Port 3000. It can be programmed to use another "working" IP address via software. If you don't know how the AP820 is configured when you first receive it, you can restore it to the default IP address by pressing the "reset IP" button on the side of the laser.

Configure the PC's TCP/IP properties as follows (assuming we are using the laser's default IP address):



Use the Demo Program included in zip files on the CD or available for download from the Acuity website. This program can be used for testing and demonstration of the general operation of the scanner. Through the demo program, one can configure scanner settings and even perform real scans and data archiving.

### Troubleshooting Notes:

If you are unable to “Connect” to the scanner through the demo program, it may be necessary to manually configure the IP:

Connect the device to a PC Within your Network Connections window, change “local area” network connection properties for the Ethernet interface in use for TCP/IP as follows:

The IP of the scanner is set to default to 192.168.1.245, the PC has to be set to the appropriate network settings to 192.168.1.244 (or a value that is compatible with the IP/mask that was last stored in the device)

Subnet mask 255.255.255.0 (or a value that is compatible with the IP/mask that was last stored in the device)

Click OK and exit out of the Network Connections windows

If necessary, reboot pc for settings to take effect.

Power up the scanner and the laser line should be visible.

Ping the IP of the scanner with the command ping 192.168.1.245-t, if the scanner responds you can start the demo software. If there is no response check the PC's IP settings again and also the Ethernet cables.

Start the demo software.

The unit is ready to use.

## 3. General Description

---

The Acuity™ AP820 Measurement Scanner is a laser-based device intended for non-contact dimension measuring and checking of surface / object profiles, shapes, target positions, displacement, dimensions, sorting and sensing. By measuring distances across a laser line, it replaces the functionality of many single-point laser sensors to achieve surface profile measurements.

The AP820 scanner series includes numerous scanners models which vary on their measuring Range, Field of View and mechanical dimensions.

### 3.1 Principles of Operation

---

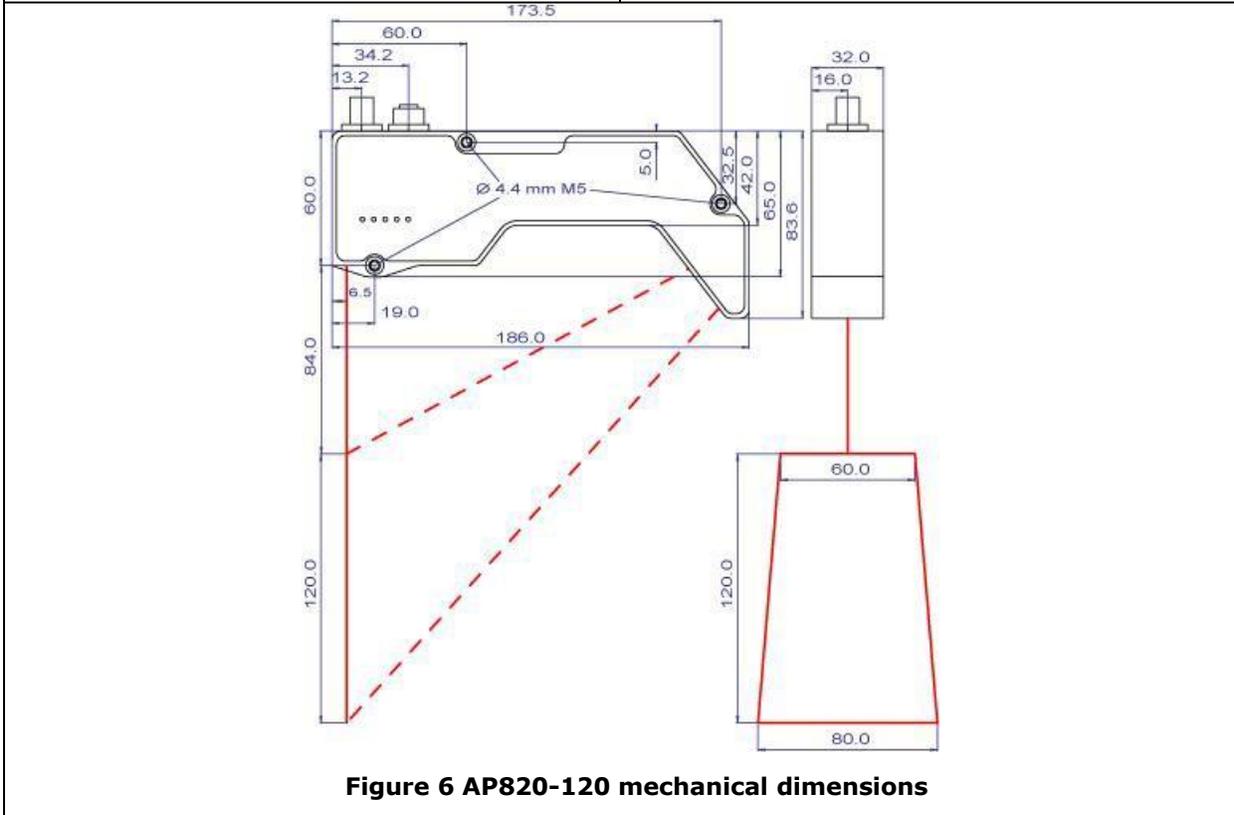
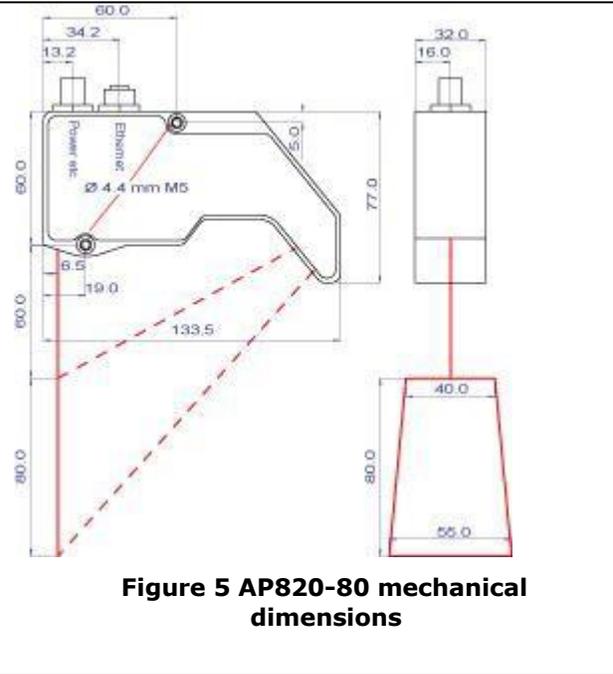
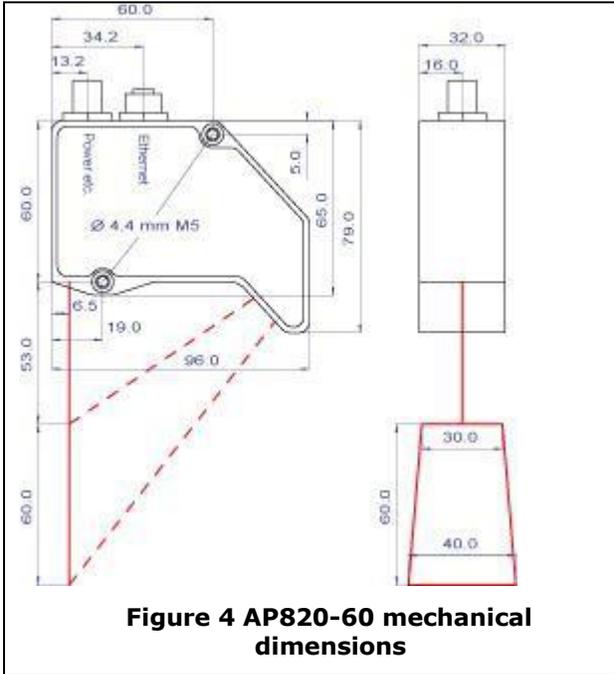
The AccuProfile™820 Profile Measurement Scanner uses optical triangulation principles to measure surface heights by projecting a beam of visible laser light that creates a line on a target surface. Reflected light from the target surface is viewed from an angle by a two-dimensional CCD detector array, 580 by 792 pixels inside the AP820 scanner. The 2D contour profile with 550 points is calculated by the scanner's signal processor from the pixel data from the diffusely reflected laser line. The height distance profile is transmitted through Ethernet communications to a PC computer. Real-time 3D profiling is created by synchronizing the position of the scanner with encoder inputs from conveyors, linear stages or robotic movements.

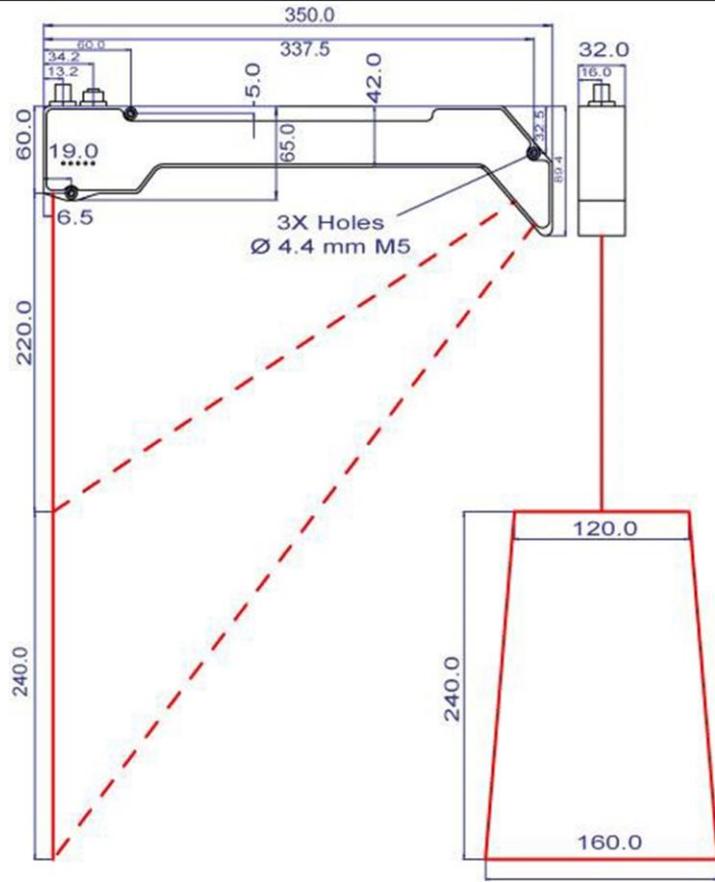
### 3.2 Mechanical Dimensions

---

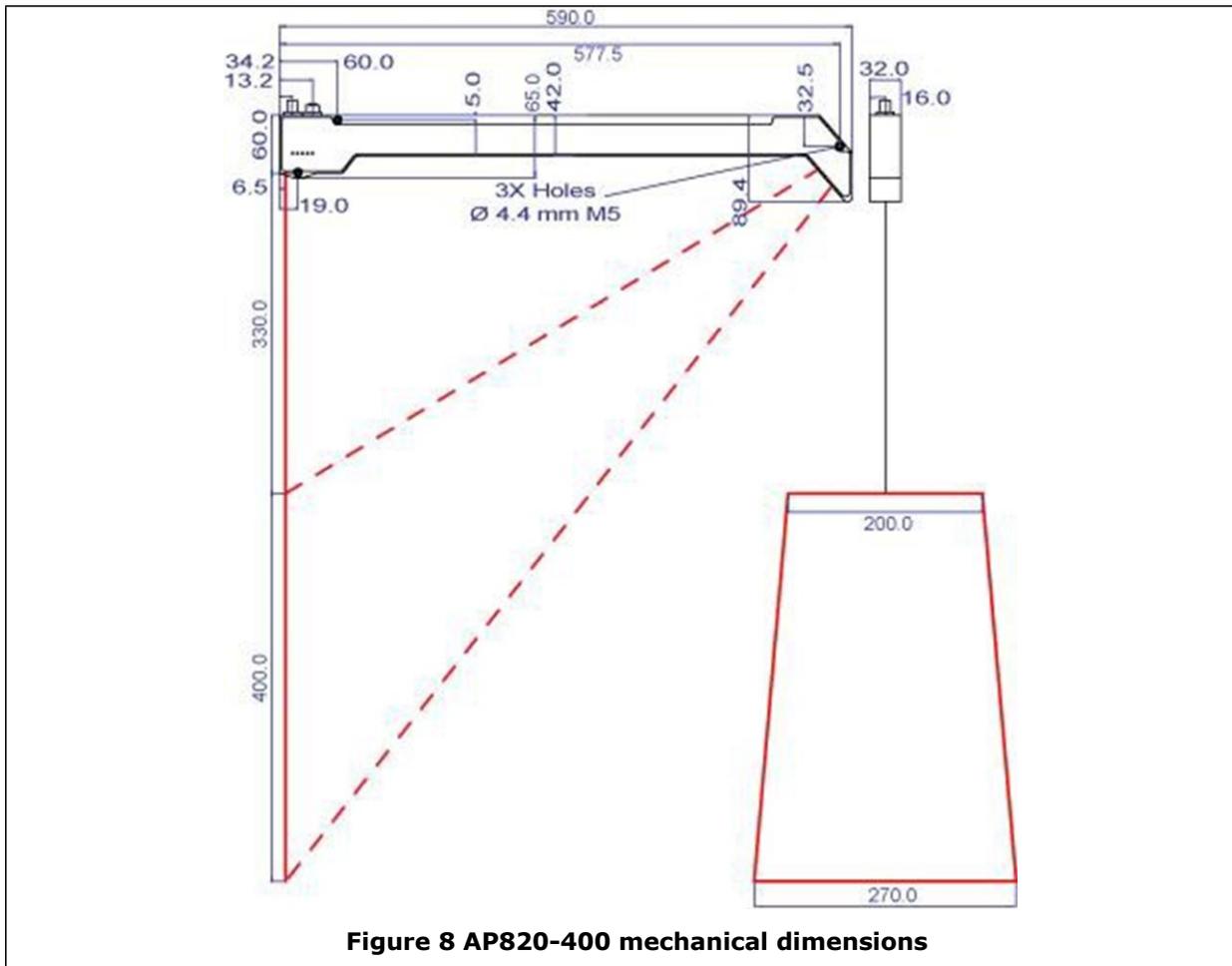
The AccuProfile™ 820 Scanner models have several different enclosure configurations and sizes.







**Figure 7 AP820-240 mechanical dimensions**



### 3.3 Contents

Your AP820 Scanner includes the following items:

1. The scanner unit
2. A 2m Ethernet cable with connector (purchased separately)
3. A 2m power / data cable with solder wire termination (purchased separately)
4. This User's Manual (part number LLL000821)

### 3.4 Laser Safety

The AP820 scanners have a visible laser line produced by a semiconductor laser diode. The laser class of the scanner depends on the model specifications, but may range from a Class 2 to a Class 3B laser device in accordance with EN60825-1:2001-11 and FDA 21CFR 1040. This laser device should not be aimed at the human eye. Installers of laser sensors should follow precautions set forth by FDA 21CFR 1040.10 or by their local safety oversight organization.

Do not direct the laser beam at other people to avoid potential eye hazards. If required, use eyewear specifically designed to block laser light of the wavelength used by the sensor.



Figure 9 AP820 laser labels

### 3.5 Sensor Maintenance

---

The AP820 sensor requires little maintenance from the user. The sensor lenses should be kept clean of dust buildup as a part of regular preventative maintenance. Use compressed air to blow dirt off the window then use delicate tissue wipes with a light solvent such as isopropyl alcohol or water. Avoid using pressurized water and do not use abrasive wipes on the optical glass. For heavy dust or dirt buildup on the lenses, rinse the lenses before wiping clean to avoid lens scratching. If your sensor does not function according to specifications, contact Schmitt Industries, Inc.

### 3.6 Sensor Service

---

The AP820 sensor has no user-serviceable parts. Do not open the unit. Refer all service questions to Schmitt Industries, Inc.

### 3.7 Sensor Specifications

---

Go to: <http://www.acuitylaser.com/docs/ap820-data-sheet.pdf>

## 4. Installation and Checkout

---

### 4.1 Mounting

---

The scanner should be securely mounted to rigid heat dissipating mechanical fixtures, brackets or stages. In environments of high vibration, consider using insulating bushings with a bracket to minimize the transfer of this movement to the scanner. The floors of some industrial environments could transfer vibration to scanner fixtures and care should be taken to eliminate this source of displacement measurement error.

**Never weld the scanner in place and avoid using a welder's arc on metal fixtures with the scanner connected.**

Consult the appropriate mechanical drawings (see section Mechanical Dimensions 3.2) to locate the size and location of the mounting holes for each scanner model. For the smaller models, the mounting holes are located on the side of the scanner head to conserve space. The small enclosures have two mounting holes on the side of the sensor. The medium enclosure models have three through holes that should be used for secure mounting.

Sensors should be mounted perpendicular to the target surface that it will be measuring.

Mount the sensor at the appropriate height above the target surface. Consult the product data sheet to determine the sensor's standoff distance. It will only measure distance between the Range Beginning and Range End.

### 4.2 Cabling

---

The AP820 sensors have two cable receptacles, one for Ethernet and one for Power / Interface. The sensors can be ordered with cables of desired length, or users may choose to manufacture their own cables.

#### 4.2.1 Control cable

---

The Control cable uses an M12-A-Coded Control and power supply plug.

Wire Color	Pin Number	Function	connectivity box label
White	1	Power +10 – 30 VDC, supply	Red
Brown	2	Digital Input 1 / Position indicator	Not connected
Green	3	Ground or 0 V	Black
Yellow	4	Digital Input 2 / Position indicator	Not connected
Gray	5	Sync OUT	Pink
Pink	6	Sync IN / Hardware Trigger	White
Blue	7	TxD	Green
Red	8	RxD	Yellow
Screen		Tied to connector plug housing	

The AP820 scanner's OEM multipurpose cable has solder tail wires. Connection and termination according to the instructions is essential for correct sensor operation. Read the wire descriptions for connection information. The color labels on the connectivity box are different from the AP820 wire colors and are shown above.

### 4.2.1.1 Power Supply

---

The Green wire is the Power Supply Common return, also named Ground. It carries the return current for the power supply.

The White wire is the Power Supply Input to the sensor. The sensor accepts a wide range from +10 to +30 VDC power and consumes <4 W.

Power supplies from 10 VDC to 30 VDC may be used. Higher voltages will result in excessive current drawn by the over voltage protection circuitry and may cause permanent damage.

### 4.2.1.2 Drain Wire (Clear)

---

The cable may have an un-insulated wire, drain, that connects the stainless steel harness plug of the cable. It should also be connected to ground at the power supply end of the cable.

## 4.2.2 Ethernet cable

---

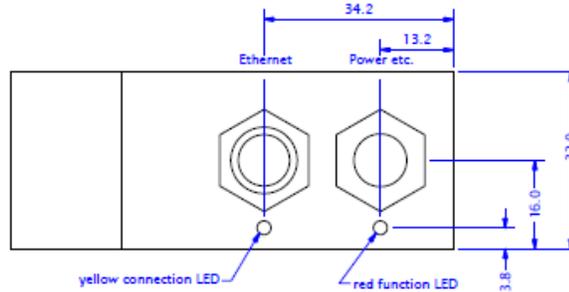
The Ethernet cable uses a M12-D-Coded 4pol Industrial Ethernet cable. The other end is a standard RJ45 connector.

Connection	Pin-No.	Function	Cable colors	Note
M12 round 4-pol D-coded female	1	Tx+	green + white	Sending data Ethernet +
	2	Rx+	red + white	Receiving data Ethernet +
	3	Tx-	green	Sending data Ethernet -
	4	Rx-	red	Receiving data Ethernet -
	Screen			Connected with the connector plug housing

## 4.3 Power On

Caution: be sure that the laser will not cause an eye hazard.

When power is applied the red or blue laser line will be emitted from the top window of the AP820 scanner. The LEDs adjacent to the cable connectors on the scanner head should illuminate.



**Figure 10 LED Configuration**

Status LEDs	Meaning	Color	OK when
System OK	Illuminates after self-check	Red	Flashing
LAN active	Ethernet Link sending data	Yellow	Flashing

### 4.3.1 Communications Check

To begin communication with the scanner, you may use your own application software or the free, demo software described in section 2.1.3.

### 4.3.2 Verifying Operation

With the scanner mounted above the target surface or reference surface, adjust its height so that the surface is slightly closer than the *Range End* for the particular sensor model you are operating. See the data sheet to verify the *Range End* for your model.

With the demo software open and working, place an opaque object of known height onto the reference surface. Zoom the image if necessary and verify that the measured height of the object is equal to (within specification tolerances) the known object height.

## 5. Ethernet Interface Operation

---

An Ethernet interface will be used for the transmission of the profile data. The recommended procedure for data readout is described in the programming example. For a normal operation of the scanner it is not necessary to send initialization data to the scanner. The electronics in the head of the scanner will store all settings automatically in order to provide the optimal profile. Only in cases where the object has a very strong color / reflectivity variation surface is it useful to activate the AGC function, as described in the register of functions.

The values in this description that are represented in hexadecimal format will be represented in C++ format. Therefore 0x23 will be the hexadecimal value 23.

### 5.1 Changing the IP-address

---

The AP820 scanners have a *default* IP-address, as well as a *working* IP-address. See the table in section 6.1 for details. Changes to the *working* IP-address will be saved permanently.

The default IP-address will be activated by the **RESET IP** button.

The default IP-address is fixed and cannot be changed by the user.

Another working IP-address can be set through the Web-Server. The procedure is explained in section 10.1.

## 6. Transmission Formats Sensor Data

Blue marked words refer to the description of data formats.

Addresses without a corresponding headword in the register "Description" do not contain any function but can contain data.

### 6.1 Scanner data

The profile data as well as the status information will be transmitted at 2048 Bytes per Scan. The composition of the packet is described in the following table.

Address	Bytes	Description	Meaning	Factory setting	
00 to 05	[6]	Default MAC-address		00:08:DC:00:00:00	Status information 1
06 to 09	[4]				
10 to 13	[4]	Default GateWay		192.168.1.	
14 to 17	[4]	Default Subnet Mask		255.255.255.0	
18 to 21	[4]	Default IP-address		192.168.1.245	
22 to 23	[2]	Default TCP-Port		3000	
24 to 25	[2]				
26 to 31	[6]	Working MAC-address	xx:xx:xx = from serial number	00:08:DC:xx:xx:xx	
32 to 35	[4]				
36 to 39	[4]	Working GateWay		variable	
40 to 43	[4]	Working Subnet Mask		variable	
44 to 47	[4]	Working IP-address		variable	
48 to 49	[2]	Working TCP-Port		variable	
50 to 61	[12]				
62	[1]	Image number	continuously 0 - 253		
63 to 65	[3]				
66 to 1515	[1450]	Profile data	1450 Bytes, which is 290 sets of 5 bytes per point		
1516 to 1524	[9]				Status information 2
1525 to 1528	[4]	Generator data for position			
1529	[1]				
1530 to 1560	[31]	Register of functions	Readout of the register of functions	Byte 0 to 30	
1561	[1]				
1562	[1]	Sensor temperature			
1563	[1]	Scanner Status Register			
1564	[1]	Version evaluation Reg1			
1565	[1]				
1566 to 1570	[6]				
1571 to 1573	[3]	Switch-on Counter			
1574	[1]	Inputs		Bit 0 for input 1, Bit 1 for input 2	

1575	[1]	Outlet of laser adjustment control		
1576	[1]	Version evaluation Reg2		
1577 to 2040	[464]			
2041 to 2042	[2]	Sensor Pixel amount horizontal		Typ. = 290
2043 to 2044	[2]	Sensor Pixel amount vertical		Typ. = 752
2045 to 2047	[3]	FIFO-Charging level	maximum value 524287	

## 6.2 Info-Telegram

The Info-Telegram can be received by sending the command 0x21 to the scanner. The packet described below will be received with 2048 Bytes. In the field "Status information 1" on address 60 will be the value 0x10, in order to identify the Info-Telegram. The field "Status information 1" is not described here specifically, because it is composed exactly like the scanner data composition. Sending 0x21 during a measurement is not recommended. It will slow down the scan.

Address	Bytes	Description	Meaning	Factory setting
00 to 65	[66]	Status information 1	See Description Scanner data	
66	[1]	Sensor temperature		
67	[1]	Scanner Status Register		
68	[1]	Version evaluation Reg1		
69	[1]			
70 to 74	[5]	Operating hours counter		
75 to 77	[3]	Switch-on counter		
78	[1]	Inputs		
79	[1]	Outlet of laser adjustment control		
80	[1]	Version evaluation Reg2		
81 to 97	[17]			
98 to 99	[2]	Sensor Pixel amount horizontal		
100 to 101	[2]	Sensor Pixel amount vertical		
102 to 105	[4]	Serial number		
106 to 107	[2]	Z Start measurement range		
108 to 109	[2]	Z measurement range		
110 to 111	[2]	X Scan range AMB	At the beginning of the measurement range	
112 to 113	[2]	X Scan range EMB	At the end of the measurement range	
114 to 125	[12]			
126	[1]	Sensor Data	Sensor position, mirror, Data resolution	
127 to 129	[3]			
130 to 135	[6]	Version Firmware	ASCII String	e.g. 2.0.60
136 to 137	[2]			
138 to 168	[31]	Register of functions	Readout of the register of functions	Byte 0 to 30
169	[1]			
170 to 172	[3]	FIFO-Charging level	maximum value 524287	
173 to 2047	[1875]			

## 7. Data Formats

---

### 7.1 Profile data

---

For every profile, 290 points will be sent. Every point in the profile will be defined through an X- and a Z-value. The I-value refers to the intensity of a point and can be used to define points as valid or invalid.

The scanner provides linearized profile data by default.

Switch to non-linearized profile data with the commands 0x16 0x80.

Calculate a linearized point from the X- and Z-value as follow:

$$X = \frac{X.value[13-0]}{X.max} * X \text{ Scan Range EMB} * \text{Data resolution} \quad [\text{mm}], X\text{-max} = 4096$$

$$Z = \frac{Z.value[13-0]}{Z.max} * Z \text{ Measurement Range EMB} * \text{Data resolution} \quad [\text{mm}], Z\text{-max} = 4096$$

The values X-max, Z-max, X Scan range EMB, Z Measurement range and Data resolution can be received with the Info-Telegram. Data resolution is 0.1 for all scanners except the AP820-1000 where it is 1.0.

Byte Number	Bit No.	Meaning
Profile Reg.1	6 to 0	X-value Bit 6 to 0
Profile Reg.2	6 to 0	X-value Bit 13 to 7
Profile Reg.3	6 to 0	Z-value Bit 6 to 0
Profile Reg.4	6 to 0	Z-value Bit 13 to 7
Profile Reg.5	7 to 0	I-value Bit 7 to 0 value range = 0x01(dark) to 0xFE(light)

## 7.2 Encoder Position Data

---

The data for encoder positions are only available when A and B phases of an incremental encoder are connected to the digital inputs 1 and 2.

Encoder inputs:

Protocol Incremental Signal A+B 90° Phase shifted

Input level Low = 0 ... 2 V High = 5 ... 30 V

Register	Bit No.	Meaning
Position generator Reg.1; Byte 1525	6 to 0	Position generator Bit 6 to 0
Position generator Reg.2; Byte 1526	6 to 0	Position generator Bit 13 to 7
Position generator Reg.3; Byte 1527	6 to 0	Position generator Bit 20 to 14
Position generator Reg.4; Byte 1528	5 to 0 6	Position generator Bit 26 to 21 Direction of movement of Position generator

## 7.3 Register of Functions

You can read back the data of the register of functions of the AP820 laser sensor. The content of this register is described in Section 8.2.

## 7.4 Sensor temperature

A temperature probe is built into the AP820 sensor. The temperature will be measured one time per second.

Register	Bit No.	Meaning	
Temperature Reg.1 Byte 1562	6 to 0	<b>Temperature</b>	<b>Value</b>
		+126 C°	0x7E
		+ 85 C°	0x55
		+ 25 C°	0x19
		+ 0 C°	0x00
		- 1 C°	0xFF
		- 25 C°	0xE7
Signed byte representing temperature in °C, valid range from -55 to 126° Bit 7 is the leading sign			

## 7.5 Scanner Status Register

Register	Bit No.	Meaning	Value = 0	Value = 1
Status Reg.1	Byte 1563, Bit 0			
	Byte 1563, Bit 1	Register condition	After reset	If changed
	Byte 1563, Bit 2			
	Byte 1563, Bit 3	Laser	on	off
	Byte 1563, Bit 4	Measurement control	Continuous mode	Single measurement with Trigger
	Byte 1563, Bit 5	Laser control	automatic	manual

## 7.6 Version Evaluation Registers

The version of the evaluation will be generated from the registers 1 + 2. Register 2 shows the third position of the version number.

Register	Bit No.	Meaning
Version Reg.1; Byte 1564	7 to 0	Value 36 corresponds to e.g. version 3.6
Version Reg.2; Byte 1576	7 to 0	Add to the content Reg.2 with e.g. content 1 results version 3.61

## 7.7 Operating Hours Counter

The counter increases every 250 msec. For the result in seconds, the operating hours counter [31 to 0] has to be divided into four.

Register	Bit No.	Meaning
Reg.1; Byte 1566	6 to 0	operating hours counter Bit 6 to 0

Reg.2; Byte 1567	6 to 0	operating hours counter Bit 13 to 7
Reg.3; Byte 1568	6 to 0	operating hours counter Bit 20 to 14
Reg.4; Byte 1569	6 to 0	operating hours counter Bit 27 to 21
Reg.5; Byte 1570	6 to 0	operating hours counter Bit 31 to 28

## 7.8 Switch-on Counter

Every time the sensor is turned on, the value will increase to 1.

Register	Bit No.	Meaning
Reg.1; Byte 1571	6 to 0	Switch-on counter Bit 6 to 0
Reg.2; Byte 1572	6 to 0	Switch-on counter Bit 13 to 7
Reg.3; Byte 1573	2 to 0	Switch-on counter Bit 16 to 14

## 7.9 Input Status

This reports the condition of the inputs 1 and 2.

Register	Bit No.	Meaning
Input register.1; Byte 1574	0	Input 1
	1	Input 2

## 7.10 Adjusting the laser output

This is the value which the laser power adjustment control was on during *automatic* operation mode.

To manually adjust this value, aim the scanner to the surface to be scanned. Continue by reading the value of the laser output control. Now switch the laser adjustment control to *manual* operation mode (see section 8 for how to change values) and write this value (Output of the adjustment control \* 4) into the register "Exposure Control". The laser output adjustment control is now adjusted on the object to be measured.

Register	Bit No.	Meaning
Laser adjustment control Reg.1; Byte 1575	7 to 0	0x00 = maximum exposure time and brightness. 0xFF = minimum exposure time and brightness.

## 7.11 Number of Horizontal and Vertical Pixels

Half of the number of pixels of the sensor in X-direction is indicated as the value "Sensor Pixel amount horizontal".

Register	Bit No.	Meaning
Number of Horizontal pixels Reg.1; Byte 2041	6 to 0	Sensor Number of pixels Bit 6 to 0
Number of Horizontal pixels Reg.2; Byte 2042	6 to 0	Sensor Number of pixels Bit 13 to 7

Register	Bit No.	Meaning
Number of Vertical pixels Reg.1; Byte 2043	6 to 0	Sensor Number of pixels Bit 6 to 0
Number of Vertical pixels Reg.2; Byte 2044	6 to 0	Sensor Number of pixels Bit 13 to 7

## 7.12 FIFO-First In, First Out

---

The Scanner has a Data FIFO of 512 kByte. This value indicates the charging level.

Register	Bit No.	Meaning
Reg.1; Byte 2045	7 to 0	FIFO- charging level Bit 7 to 0
Reg.2; Byte 2046	7 to 0	FIFO- charging level Bit 15 to 8
Reg.3; Byte 2047	2 to 0	FIFO- charging level Bit 18 to 16

## 7.13 Serial number

---

The reading of the serial number Bit 27 to 0, results in a decimal number with 7 digits.

For example, 1009003 Starting from the left, the first two digits refer to the month on which the product was manufactured, followed by the year and a continuous 3 digits number.

Register	Bit No.	Meaning
Reg.1; Byte 102	6 to 0	serial number Bit 6 to 0
Reg.2; Byte 103	6 to 0	serial number Bit 13 to 7
Reg.3; Byte 104	6 to 0	serial number Bit 20 to 14
Reg.4; Byte 105	6 to 0	serial number Bit 27 to 21

## 7.14 Z and X values Info-Telegram address 106 to 125

---

Every one of the 4 values consists of a double register. The information in the address 106 to 113 are indicated as follows: (value \* Data resolution) mm

Register	Bit No.	Meaning
Reg.1	6 to 0	value Bit 6 to 0
Reg.2	6 to 0	value Bit 13 to 7

## 7.15 Sensor Data

---

Register	Bit No.	Meaning	value = 0	value = 1
Sensor Data Reg.1	3	Data resolution	0.1 mm	1.0 mm

## 7.16 Version Firmware

---

This is the version of the sensor's Ethernet firmware. An ASCII String will be sent.

For example see below.

Register	Bit No.	Meaning e.g. Version 06.0.2
Firmware Reg.1	7 to 0	"0"
Firmware Reg.2	7 to 0	"6"
Firmware Reg.3	7 to 0	."
Firmware Reg.4	7 to 0	"0"
Firmware Reg.5	7 to 0	."
Firmware Reg.6	7 to 0	"2"

## 8. Scanner control

### 8.1 Data format for Register-addresses and Data

For normal operation, it is not necessary to write the registers. An initialization of the scanner is not necessary.

The control commands are composed as follow:

	Bit No.	Meaning	value = 0	value = 1
control command	6 to 0	Data		
	7	Identified	Datas are address	Datas are register value

To be able to write the register value, first the register address will be sent, followed by the value. The register address that was set will be maintained until a new register address is sent.

In cases of double registers, the register value will be assumed only after the writing of a higher-quality register.

In cases of control registers in which the column "Bit No." is marked with "\*", it is sufficient to write the register address to perform the function.

Registers that are not included in the table in section 8.2, should not be addressed.

### 8.2 Register of functions

Address	Bytes	Bit No.	Description	Meaning		Default
0 to 1	[2]	6 to 0	Exposure control	Manual exposure control		0
2 to 3	[2]					
4	[1]	6 to 0	Limiting the Range	min Z		0
5	[1]	6 to 0	Limiting the Range	max Z		Sensor Pixel amount vertical/8
6 to 7	[2]	6 to 0	Amplification	Amplification grey scale image		500
8 to 10	[3]					
11	[1]	0	AGC (amplification control)	0 = off	1 = on	0
12	[1]	0	Laser on/off	0 = off	1 = on	1
13	[1]	0	Ambient light filter	0 = off	1 = on	0
14	[1]	*	Reset generator data for position	generator data for position will be set on "0"		
15	[1]	0	Synchronization Mode	0 = simultaneous	1 = alternately	0
16 to 17 & 19	[3]					0
18	[1]	0	Scanrate	0 = 195 Hz	1 = 350 Hz	0

20	[1]	3	Measurement control	0 = continuously	1 = Trigger mode	0
21	[1]	0	Exposure control Mode	0 = Automatic	1 = Manual	0
22	[1]					
23 to 27	[5]					
28	[1]	*	Reset FIFO	Erases all data in the internal FIFO		
29	[1]	*	Software Trigger	Function only when measurement control = 1		
30	[1]	*	Reset Sensor	All register values on Default value		
31	[1]	*	Reset Ethernet Module			
32	[1]					
33	[1]	*	Request Info-Telegram			
34 to 127	[94]					

## 9. Description of the control commands

### 9.1 Exposure control

This register is for the manual control of the sensor's exposure. It only has an effect when the register exposure control mode is set to 1 for manual.

Register	Bit No.	Meaning
Reg.1	6 to 0	value Bit 6 to 0
Reg.2	2 to 0	value Bit 9 to 7

0x000 = maximum exposure time and brightness.  
0x3FF = minimum exposure time and brightness.

### 9.2 Calculation of the Exposure Time

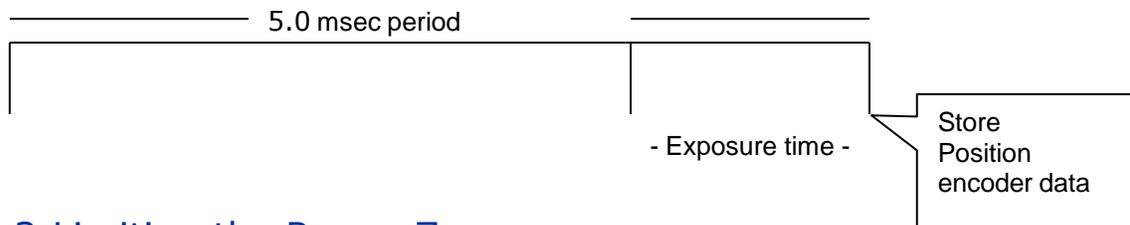
The sensor works with 200 half images per second. This is a 5.0 msec period.

The maximum exposure time is 1.0 msec. The exposure time can be keyed in 1024 steps.

Based on this, the minimum exposure time can be calculated as:

maximum exposure time/1024=1 µsec

To identify from which position on the exposure was carried out, it is required that a position encoder is plugged to the scanner. The position will be saved at the end of an image.



### 9.3 Limiting the Range Z

The registers for limiting the range Z define the area in which the sensor is being read out. A profile cannot be generated outside of a defined window.

With this function, it is possible to put a measurement "window" in front of the sensor, in order to reduce the Z-measurement range at the front and back.

### 9.4 Amplification

The amplification of the sensor signal can be influenced. In cases of very dark and reflecting surfaces, the value can be increased here.

Register	Bit No.	Meaning
Reg.1	6 to 0	value Bit 6 to 0
Reg.2	2 to 0	value Bit 9 to 7

0x000 = minimum amplification.  
0x3FF = maximum amplification.

## 9.5 AGC (amplification control)

---

When this function is activated, the scanner dynamically controls the amplification of the sensor signal.

This is an automatic function that offers optimal results on most surfaces, in cases of surfaces with varying reflectivity. However, when the scanner looks into "emptiness" with no target within range, the sensor is sensitive to ambient light.

## 9.6 Ambient Light Filter

---

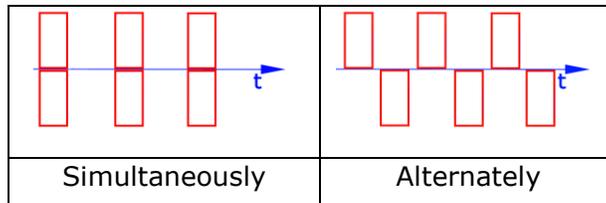
The ambient light filter is helpful for measurements in very bright ambient (sun glare) and in the vicinity of welding arcs. This bandpass filter is especially helpful on highly- reflective surfaces, like metals. By using this filter, the resolution of the scanner will be slightly diminished.

## 9.7 Synchronization Mode

---

This mode controls how two scanners work together when connected with the help of synchronization inputs and outputs.

The function "alternately" permits the use of two scanners that may interfere (laser of sensor 1 seen by detector of sensor 2 and vice versa) with each other. This way, it is possible to measure the very same spot on a target with both scanners, doubling the effective measuring rate.



Set up this function by connecting the Sync-Out outlet of scanner 1 with the Sync-In input of scanner 2. The GND Pins of both scanners also must be connected with each other. The external Trigger function is not available, when the scanners are being synchronized with each other. In this mode the image counters of the scanners run synchronously as well. This makes it possible to tell which pictures were taken simultaneously by comparing the numbers of the images.

## 9.8 Measurement Control Trigger

---

This function controls the Trigger Mode.

Scans or images can be obtained only when an external hardware trigger has been released over a Pin Sync-In or when a software trigger has been released with a register 29. After the trigger two scans or two grey scale images are sent.

A "Hardware-Trigger" will be released with a 0/1-Slope on the Sync-In input. The signal should be sufficiently bounce-free.

During a running measurement it will not be possible to release a further measurement and the command is ignored. The maximum image-frequency in this mode is about 60 Hz.

# 10. Description of the Web-Servers

Address the scanner by using the integrated Web-Server with the help of a web-browser. The parameters and the scan profile will be indicated.  
A scan profile update is not performed automatically until the scanner is updated via the website.

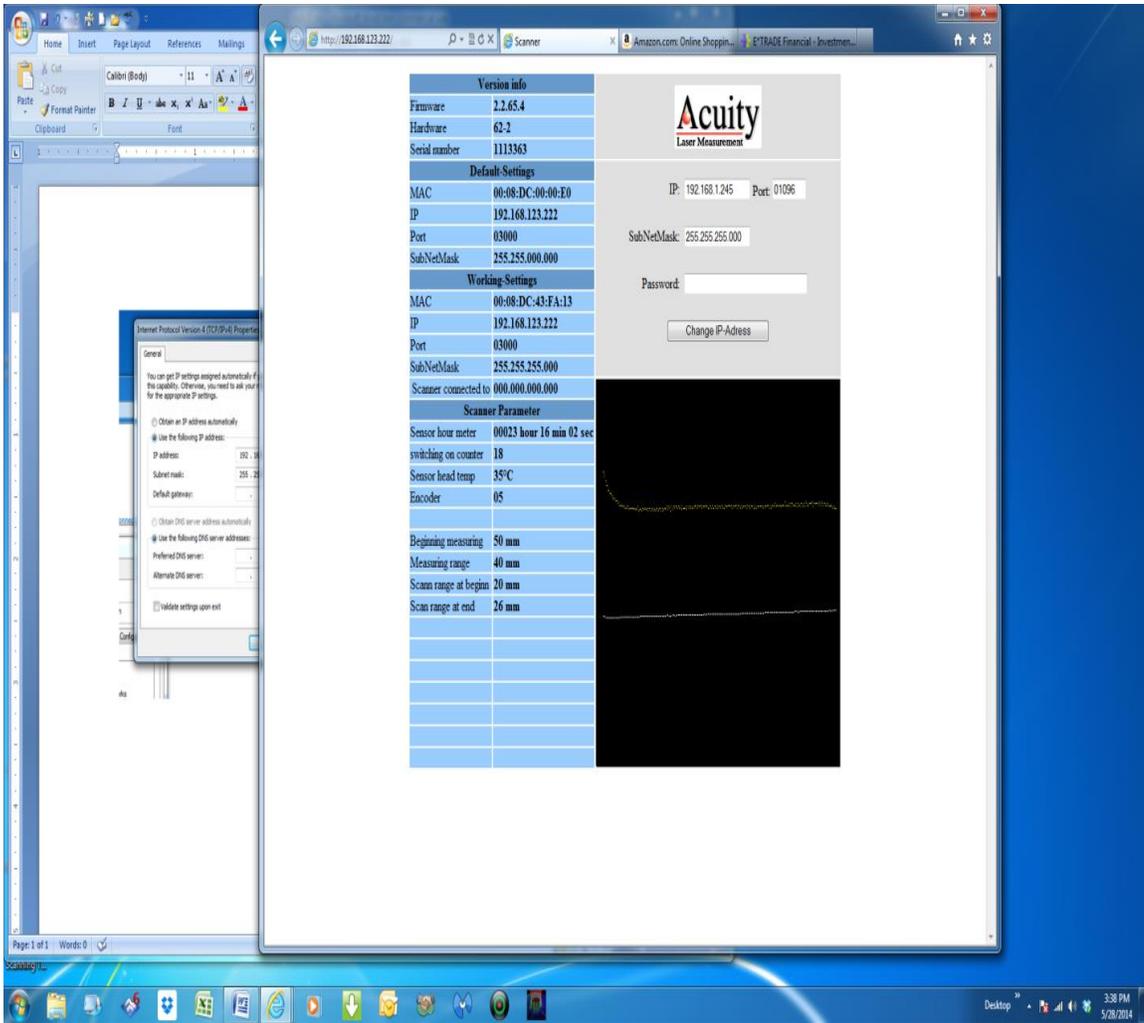
Additionally, the access with the web-browser allows the set up directly of the working IP-address, the Port and the Subnet mask.

## 10.1 Changing the IP-address

Type the new IP, port and subnet mask into the appropriate fields of the web browser.

The password is: **q4**

After pressing "Change IP-address", the scanner confirms the new data on a report within a new screen page. A new scan profile is also presented.



## 11. Programming using Windows DLL Calls

---

The following discussion is an attempt to show, step by step, what is required to set up, program, and get results from an AP820 laser using Windows DLL function calls. To illustrate the use of the Windows DLL functions, this document presents examples of how to call each of the functions from the **Visual Basic DotNet** environment. Visual Basic DotNet was selected because it is the most straightforward to use of the Windows programming languages. Experienced C++ or C# programmers should not have difficulty translating the methodology into their programming language. Note the variation between C++ and VB hexadecimal expressions. The discussion below assumes familiarity with Visual Basic DotNet, and associated terms and concepts such as managed and unmanaged code, etc.

### 11.1 Importing EthernetScanner.dll

---

The functions that access the AP820 are exported in a library called EthernetScanner.dll. These functions are:

- EthernetScanner\_Connect
- EthernetScanner\_Disconnect
- EthernetScanner\_GetConnectStatus
- EthernetScanner\_GetScanRawData
- EthernetScanner\_GetInfo
- EthernetScanner\_WriteData
- EthernetScanner\_GetVersion

We will import these into VB.net using the DllImport attribute in the System.Runtime.InteropServices namespace, which we will reference at the top of the module containing the code to import the dll functions.

```
Imports System.Runtime.InteropServices
```

Then, we use DllImport to import the functions from the dll.

```
<DllImport("EthernetScanner.dll")>
    Private Function EthernetScanner_Connect(ByVal IPAddr As String, ByVal
    Port As String, ByVal Timeout As UInteger) As IntPtr
    End Function

<DllImport("EthernetScanner.dll")>
    Private Function EthernetScanner_Disconnect(ByVal pScanner As IntPtr) As
    UInteger
    End Function

<DllImport("EthernetScanner.dll")>
    Private Sub EthernetScanner_GetConnectStatus(ByVal pScanner As IntPtr,
    ByRef Status As UInteger)
    End Sub

<DllImport("EthernetScanner.dll")>
    Private Function EthernetScanner_GetInfo(ByVal pScanner As IntPtr, ByRef
    ScanInfo As ScanInfoStruct, ByVal ScanInfoSize As UInteger, ByVal Timeout
```

```

    As UInteger) As UInteger
End Function

<DllImport("EthernetScanner.dll")> _
    Private Function EthernetScanner_WriteData(ByVal pScanner As IntPtr,
        ByVal Buffer As Byte(), ByVal BufferSize As UInteger) As UInteger
    End Function

<DllImport("EthernetScanner.dll")> _
    Private Function
        EthernetScanner_GetVersion(<MarshalAs(UnmanagedType.LPStr), Out()> ByVal
        Version As StringBuilder, ByVal Result As UInteger) As UInteger
    End Function

<DllImport("EthernetScanner.dll")> _
    Private Shared Function EthernetScanner_GetScanRawData(ByVal pScanner As
        IntPtr, ByVal Buffer As Byte(), ByVal BufferSize As UInteger, ByVal
        ScanProfileMode As UInteger, ByVal Timeout As UInteger) As Boolean
    End Function

```

We will discuss each function individually as we go. Note that, with the exception of `EthernetScanner_GetVersion`, there is a managed type with default marshalling behavior that matches the unmanaged type in the dll. The exception, `EthernetScanner_GetVersion`, requires a null-terminated string in the buffer passed to it by a pointer. To accomplish this, we will use the buffer of the `StringBuilder` type and the `MarshalAs` attribute in `System.Runtime.InteropServices` to marshal the `StringBuilder` as a pointer to a null-terminated string.

## 11.2 Calling `EthernetScanner_GetVersion`

---

The `EthernetScanner_GetVersion` function is a fast and easy way to prove that we are calling the dll correctly, because it is not dependent on the laser in any way. Having imported the dll using `DllImport` as in the above example, we can use the following code to call `EthernetScanner_GetVersion`:

```

Public Function GetVersion() As String
    'Buffer must be at least 256 bytes long
    Dim VersionBuffer As New System.Text.StringBuilder(256)
    Dim Result As UInteger

    Result = EthernetScanner_GetVersion(VersionBuffer, 256)
    'This function returns the length of the Version String on success
    'or -1 on Error
    If Result > -1 Then
        Return VersionBuffer.ToString
    Else
        Return "Error"
    End If
End Function

```

Note: The above function should return a string similar to: "1.0.2.1 110629".

## 11.3 Establishing Ethernet communication with the AP820

---

Refer to section 2.1.2

Later, once you've established communications with the laser, you can configure the laser's working IP address to be another address, if desired.

## 11.4 Calling EthernetScanner\_Connect, Ethernet\_Scanner\_Disconnect, and EthernetScanner\_GetConnectStatus.

---

To prove that the AP820 is connected properly and that we are able to "talk" to it via Ethernet, we will have to call EthernetScanner\_Connect, and then verify that we are connected by calling EthernetScanner\_GetConnectStatus.

The form of the function EthernetScanner\_Connect is:

```
Private Function EthernetScanner_Connect(ByVal IPAddr As String, ByVal Port As String, ByVal Timeout As UInteger) As IntPtr
```

Where:

IPAddr is a string containing the IP Address (e.g. "192.168.123.222")

Port as a string containing the TCP Port (default value is "3000")

Timeout is an unsigned integer containing the number of milliseconds to wait for a response

If successful, this function returns a pointer to an object representing the AP820. (This pointer will be required to be passed to other functions in the dll.) If this function is unsuccessful, this pointer will be a null pointer. First, we declare a pointer for the laser object:

```
Public pScanner As IntPtr = Nothing
```

Then, we connect to the object with the following code:

```
pScanner = EthernetScanner_Connect("192.168.1.245", "1096", 500)
```

```
If pScanner = Nothing Then  
    'Failure!  
Else  
    'Success!  
End If
```

If the function does not return a null pointer, we have successfully connected to the laser.

We will disconnect from the AP820 later by calling EthernetScanner\_Disconnect and passing it the pScanner pointer.

We can check the connect status by calling:

```
Private Sub EthernetScanner_GetConnectStatus(ByVal pScanner As IntPtr, ByRef Status As UInteger)
```

Where:

pScanner is the pointer returned by EthernetScanner\_Connect

and

Status is a unsigned integer used as a buffer to contain the status code when the function returns.

The status codes are:

- 0: Disconnected
- 1: Disconnecting
- 2: Connecting
- 3: Connected

Normally, when we call EthernetScanner\_GetConnectStatus after calling EthernetScanner\_Connect, we will expect the status result to be 3.

## 11.5 Getting AP820 Parameters by calling EthernetScanner\_GetInfo

---

Information about the AP820 settings can be obtained by calling EthernetScanner\_GetInfo. The information is returned in the following structure:

```
<StructLayout(LayoutKind.Sequential)> Public Structure ScanInfoStruct

    <MarshalAs(UnmanagedType.ByValTStr, sizeconst:=16)> Public
    SerialNumber As String
    <MarshalAs(UnmanagedType.ByValTStr, sizeconst:=128)> Public
    FirmwareVersion As String
    Public BeginningOfMeasuringRange As UInteger
    Public MeasuringRange As UInteger
    Public ScanRangeBeginning As UInteger
    Public ScanRangeEnd As UInteger
    Public MaxZLinearized As UInteger
    Public MaxXLinearized As UInteger
    Public MaxZNonLinearized As UInteger
    Public MaxXNonLinearized As UInteger
    Public NumZPixels As UInteger
    Public NumXPixels As UInteger
    Public ScannerTemperature As UInteger
    Public OperatingSecondsCounter As UInteger
    Public PowerCycleCounter As UInteger
    Public Fifo As UInteger
    Public PositionEncoder As UInteger
    Public PositionEncoderDirection As UInteger
    Public Protocol As UInteger
    Public Linearization As UInteger
    Public CameraMode As UInteger
    Public ProfileMode As UInteger
    Public ScannerMode As UInteger
    Public ShutterTimeManual As Integer
    Public ShutterTimeAuto As Integer
    Public PixelReadOutStart As UInteger
    Public PixelReadOutEnd As UInteger
    Public VideoGain As UInteger
```

```

Public LaserIntensityThreshold As UInteger
Public LaserTargetValue As UInteger
Public PeakWidthLimit As UInteger
Public PeakThreshold As UInteger
Public Synchronization As UInteger
Public ProtocolVersion As UInteger
Public ShutterControl As UInteger
Public Linearization2 As Integer
Public Speed As UInteger
Public FPGAVersion As UInteger
Public DigitalInput As UInteger
Public LaserValueOfProfile As UInteger
End Structure

```

To get the current settings back from the laser, pass this structure by reference to `EthernetScanner_GetInfo`:

```

Public Function GetInfo(ByRef ScanInfo As ScanInfoStruct) As Boolean
Dim Result As Boolean
Result = EthernetScanner_GetInfo(pScanner, ScanInfo, 296, 500)
.
.
.
End Function

```

## 11.6 Getting Profile Data from the AP820

---

Profile data is retrieved from the AP820 by calling `EthernetScanner_GetScanRawData`. The function is of the following form:

```

Private Function EthernetScanner_GetScanRawData(ByVal pScanner As IntPtr,
ByVal Buffer As Byte(), ByVal BufferSize As UInteger, ByVal ScanProfileMode
As UInteger, ByVal Timeout As UInteger) As Boolean

```

Where:

`pScanner` is the pointer returned by `EthernetScanner_Connect`

`Buffer` is a byte array that will contain the data (described later)

`BufferSize` contains the size of the buffer (3500)

Note that the buffer must be at least 3500 bytes, even though only the first 2048 bytes contain useful data.

`ScanProfileMode` is reserved and should always be 0

`Timeout` is the timeout time in milliseconds.

This function returns 0 on success and -1 on failure.

The profile data is returned in the byte array starting at index 66, and continuing through the next 1450 bytes to index 1515. We will describe the contents of the rest of the array later, but for now we will concentrate on these 1450 bytes containing the data.

Each successive group of 5 bytes contains:

- Byte 1 – Bits 6 to 0: Contains Bits 6 to 0 of the raw X value
- Byte 2 – Bits 6 to 0: Contains Bits 13 to 7 of the raw X value
- Byte 3 – Bits 6 to 0: Contains Bits 6 to 0 of the raw Z value
- Byte 4 – Bits 6 to 0: Contains Bits 13 to 7 of the raw Z value
- Byte 5 – Bits 7 to 0: Contains an intensity value from 1 (dark) to 254 (light)

This pattern is repeated 290 times for a total of 1450 (290 \* 5) bytes. The measured profile for a single scan is represented by 290 pixels horizontally. The laser, in fact, has 580 pixels, but only makes measurements on 290 pixels in each scan. So, for example, scan number N will return data from pixels 1, 3, 5, and so on, and the subsequent scan number N + 1 will return data from pixels 0, 2, 4 and so on. Thus, although the horizontal field of view of odd numbered and even numbered scans is the same, and the resolution of each scan is 1/290 of the horizontal field of view, the field of view is shifted by one pixel in odd numbered scans with respect to the field of view of the even numbered scan. An important consequence of this is that, if the one pixel shift of the measurement points is not important for the application, the user can treat each scan as an independent measurement and obtain 290-pixel scans at the rate of approximately 200 Hz. On the other hand, if he/she wishes to resolve the horizontal field of view into 580 parts and the measured target is assumed not to change during the time between adjacent scans, he/she can programmatically interleave the data from odd numbered scans with the data from even numbered scans and get twice the resolution (i.e. 580 pixels) at half the rate. (Example code demonstrating data "interleaving" to accomplish a 580 point profile will be presented after discussion of 290 point scanning.)

### 11.6.1 Examples EthernetScanner\_GetScanRawData

The following example illustrates the use of the EthernetScanner\_GetScanRawData function to obtain a single 290 point scan (across the field of view X):

```
Public Function GetData(ByVal XArray() As UInteger, ByVal ZArray() As
UInteger, ByVal IArray() As UInteger) As Boolean
    Dim Buffer(3500) As Byte
    Dim Result As Boolean
    Dim I As Integer

    'Size of the arrays passed to this function should be 290
    'Call EthernetScanner_GetScanRawData
    Result = EthernetScanner_GetScanRawData(pScanner, Buffer, 3500, 0, 500)
    'If everything went OK, stuff the results in the arrays
    If Result Then
        For I = 0 To 289
            '
            '           Lower 6 bits of X                               Higher 6
            bits of X
            XArray(I) = CUInt(Buffer(66 + (I * 5)) And &H7F) Or
CUInt((Buffer(66 + (I * 5) + 1) And &H7F) << 7)
            '
            '           Lower 6 bits of Z                               Higher 6 bits of
            Z
            ZArray(I) = CUInt(Buffer(66 + (I * 5) + 2) And &H7F) Or
CUInt((Buffer(66 + (I * 5) + 3) And &H7F) << 7)
            'Intensity
            IArray(I) = CUInt(Buffer(66 + (I * 5) + 4))
        Next

    End If
    Return Result
End Function
```

End Function

Note that the above function returns raw data which has not yet been scaled in mm. The full range of X is 0 to 4095, and the full range of Z is 0 to 4095. The intensity range is 1 to 254, and can be used to determine if a particular data point is valid.

To scale the data in millimeters, use the following formulae:

$$X_{mm} = (X_{raw} / 4096) * ScanRangeEnd * 0.1$$

$$Z_{mm} = (Z_{raw} / 4096) * MeasuringRange * 0.1$$

Note that ScanRangeEnd and MeasuringRange are members of the ScannerInfo structure which has previously been discussed. These ranges can be determined by calling the EthernetScanner\_GetInfo function, or alternately from the specifications for the particular model of the laser being used.

The discussion which follows explains how to interleave two "back-to-back" scans to obtain a 580 point profile (which will have the same X width but twice the resolution of the 290 point scan). Although the AP820 returns 290 points in each scan, it actually has twice that many pixels (580) and in one scan returns pixels 0, 2, 4, and so on, and the next scan returns pixels 1, 3, 5 and so on. Data from these sequential scans can therefore be interleaved to form one 580-point scan at pixels 0, 1, 2, 3, 4, 5, and so on. To assist in this, the structure returned by EthernetScanner\_GetScanRawData contains an image counter at index 62. This image counter increments by one each time a new scan is obtained, and resets to zero every 254 scans. This counter can be used to identify odd numbered vs. even numbered scans. We would use the image counter to determine whether the scan is odd or even-numbered, and whether two scans are sequential. We obtain the odd-numbered scan first, containing even-numbered pixels 0, 2, 4 and so on, then get the even-number scan second, containing the odd-numbered pixels 1, 3, 5, and so on. Whether used for interleaving sequential scans, or in applications that use each 290-point scan individually, the image counter can be used in any application to insure that, in transferring a series of scans from the laser, a scan has not been missed.

The following example interleaves two successive scans into a single 580 point profile.

```
Public Shared Function GetInterleavedData(ByVal XArray() As UInteger, ByVal Zarray() As UInteger,
ByVal IArray() As UInteger) As Boolean
    Dim Buffer(3500) As Byte
    Dim Result As Boolean
    Dim I As Integer
    Dim ImageNumber, HalfImageNumber, SecondImageNumber As Byte

    'Size of the arrays passed to this function should be 580

    'Call Ethernet_GetScanRawData until we get an odd image count first
    Do
        Result = EthernetScanner_GetScanRawData(pScanner, Buffer, 3500, 0, 500)

        If Result Then
            'Get the Image Number
            ImageNumber = Buffer(62)
            HalfImageNumber = ImageNumber / 2
        Else
            Return False
        End If
    Loop Until (ImageNumber <> HalfImageNumber * 2)

    'Got the first scan, stuff it into EVEN numbers of the array
    For I = 0 To 289
        '                Lower 6 bits of X                Higher 6 bits of X
```

```

        XArray(I * 2) = CUInt(Buffer(66 + (I * 5)) And &H7F) Or CUInt((Buffer(66 + (I * 5) + 1)
And &H7F) << 7)
        '           Lower 6 bits of Z           Higher 6 bits of Z
        Zarray(I * 2) = CUInt(Buffer(66 + (I * 5) + 2) And &H7F) Or CUInt((Buffer(66 + (I * 5) +
3) And &H7F) << 7)
        'Intensity
        IArray(I * 2) = CUInt(Buffer(66 + (I * 5) + 4))
    Next

    'Call Ethernet_GetScanRawData again
    Result = EthernetScanner_GetScanRawData(pScanner, Buffer, 3500, 0, 500)

    If Result Then
        'check that the image number is correct (one more than the first)
        SecondImageNumber = Buffer(62)
        If SecondImageNumber = ImageNumber + 1 Then
            'Got the second scan, stuff it into ODD number of the array
            For I = 0 To 289
                '           Lower 6 bits of X           Higher 6 bits of X
                XArray(I * 2 + 1) = CUInt(Buffer(66 + (I * 5)) And &H7F) Or CUInt((Buffer(66 + (I
* 5) + 1) And &H7F) << 7)
                '           Lower 6 bits of Z           Higher 6 bits of Z
                Zarray(I * 2 + 1) = CUInt(Buffer(66 + (I * 5) + 2) And &H7F) Or CUInt((Buffer(66
+ (I * 5) + 3) And &H7F) << 7)
                'Intensity
                IArray(I * 2 + 1) = CUInt(Buffer(66 + (I * 5) + 4))
            Next
        Else
            'something went wrong!
            Return False
        End If
    Else
        Return False
    End If
    Return Result

End Function

```

At this point, both the X and Z arrays will contain the 580 “interleaved” points acquired by the two “back-to-back” scans.

## 11.7 Description of other parameters returned by EthernetScanner\_GetScanRawData

---

In addition to the profile data discussed above, the buffer returned by EthernetScanner\_GetScanRawData contains the following additional information:

The buffer returned by EthernetScanner\_GetScanRawData is a byte array declared as follows:

```
Dim Buffer(3500) as Byte
```

The various parameters and their location within the byte array are in the chart in section 6.1. Explanations of these parameters can be found through section 7.

## 11.8 Writing Commands to the AP820

---

The EthernetScanner\_WriteData function is used to write an array of bytes to the laser in order to change settings or issue laser commands (such as a trigger).

The form of the function is:

```
Private Function EthernetScanner_WriteData(ByVal pScanner As IntPtr, ByVal Buffer As Byte(), ByVal BufferSize As UInteger) As UInteger
```

Where:

pScanner is the pointer returned by EthernetScanner\_Connect

Buffer contains the array of bytes to be sent to the laser

BufferSize is the number of bytes to be sent (i.e. the size of Buffer)

This function returns the number of bytes written on success, and -1 on failure.

The byte array contains two types of bytes; Register Address bytes and Register Data Values. The order of the bytes in the array must be Register Address first, followed by any data value bytes which may be required for the particular register.

A byte indicating which Register Address is to be written to must have a 0 in bit 7, and the Register Address in bits 6 to 0.

A byte indicating what data value to write to the specific register must have a 1 in bit 7, and the data value in bits 6 to 0.

## 11.8.1 Examples: Laser on, Triggering

---

In the following section, we demonstrate how to execute two laser commands (turning the laser on, and changing the triggering method), as well as how to change a setting (the Z measurement range). Once these three examples are understood, coding of any other register related operation should be straight forward by analogy.

For example, let's consider commands to turn on and to turn off the laser programmatically. We see in the Register of Functions that the laser on/off function is at Register Address 12, and that a data value of 0 in bit 0 of the register turns the laser off, and a data value of 1 in bit 0 turns the laser on.

We need a 2-byte array perform these tasks, the first byte will be the Register Address and the second byte will be the Data Value for the Register (i.e. 0 for off or 1 for on). So we might write the following code to turn the laser off:

```
Private Function TurnOffLaser() As UInteger
    Dim Buffer(1) As Byte 'Note, VB always adds 1 to the size
    Dim BytesWritten As UInteger

    ' The first byte will set the Register Address to 12
    Buffer(0) = &H0C 'bit 7 is 0, bits 6 to 0 will add to 12.
    ' The second byte will set the data value for the register to 0
    Buffer(1) = &H80 'bit 7 is 1, bit 0 is 0
    return EthernetScanner_WriteData(pScanner, Buffer, 2)
End Function
```

Likewise, to turn the laser back on we would set that register to 1 by putting &H81 in Buffer(1).

For another example, we will switch the AP820 from continuous sampling into trigger mode, in which it will require a software trigger to sample. This feature is controlled by Register Address 20, bit 3, in the Register of Functions. A "0" in bit 3 puts the laser in continuous mode, and a "1" in bit 3 puts it in trigger mode:

```

Private Function SetLaserMeasurementControlMode(ByVal SampleContinuously As
Boolean) As UInteger
    Dim Buffer(1) As Byte

    Buffer(0) = &H14 'bit 7 is 0, bits 6 to 0 indicate address 20
    If SampleContinuously Then
        'continuous mode
        Buffer(1) = &H80 'bit 7 is 1, bit 3 is 0
    Else
        'trigger mode
        Buffer(1) = &H88 'bit 7 is 1, bit 3 is 1
    End If
    Return EthernetScanner_WriteData(pScanner, Buffer, 2)
End Function

```

In trigger mode, simply writing to Register Address 29 issues a software trigger to the AP820:

```

Private Function SendSoftwareTrigger() As UInteger
    Dim Buffer(1) As Byte

    Buffer(0) = &H1D 'bit 7 is 0, bits 6 to 0 indicate address 29
    Buffer(1) = 0 'Can be anything

    Return EthernetScanner_WriteData(pScanner, Buffer, 2)
End Function

```

## 11.8.2 Example: Limiting the Range Z

---

A final example of using the EthernetScanner\_WriteData function to change laser settings demonstrates how to limit the range Z of the AP820. (Limiting the range Z of the laser may be very helpful in removing foreground or background objects from the measurement process, in order to only acquire profile data from objects of interest.) The minimum Z is set by writing to Register Address 4 and the maximum Z set by writing to Register Address 5. The units for these Z values are in Pixels \* 8, which means that each count represents 8 pixels. So, for example, the maximum Z for model AP820-20 is 752, so to set the maximum Z to half the distance (376), write 47 to Register Address 5 (i.e.  $47 * 8 = 376$ ). The following example illustrates the methodology:

```

Private Function SetMaxFieldOfView(ByVal Pixels As UInteger) As UInteger
    Dim Buffer(1) As Byte
    Dim PixelArg As Byte

    Buffer(0) = &H5 'bit 7 is 0, bits 6 to 0 are 5
    'generate the argument for the correct number of pixels
    'which is one eighth the number of pixels
    PixelArg = CByte(Pixels / 8)
    Buffer(1) = &H80 And PixelArg 'put a 1 in bit 7 and number of pixels in
bits 6-0

    Return AP820Class.WriteData(Buffer, 2)
End Function

```

## 11.9 Description of Register Addresses accessible via EthernetScanner\_WriteData

---

The following is a table of settings/commands and associated Register Addresses which can be accessed via the EthernetScanner\_WriteData function (Refer to section 9 for descriptions of the purpose of each of the commands):

Bytes 0 to 1 – Exposure Control: Maximum exposure time and brightness at &H000, Minimum exposure time and brightness at &H3FF