Productivity Open

TOPCO
It's all about control

Open-Source Agility Meets Industrial-Grade Toughness

Industrial-Grade
Arduino-Compatible CPU

P1AM-100 CPU

C++ UL-certified CPU that combines open source control with AutomationDirect industrial I/O

C++

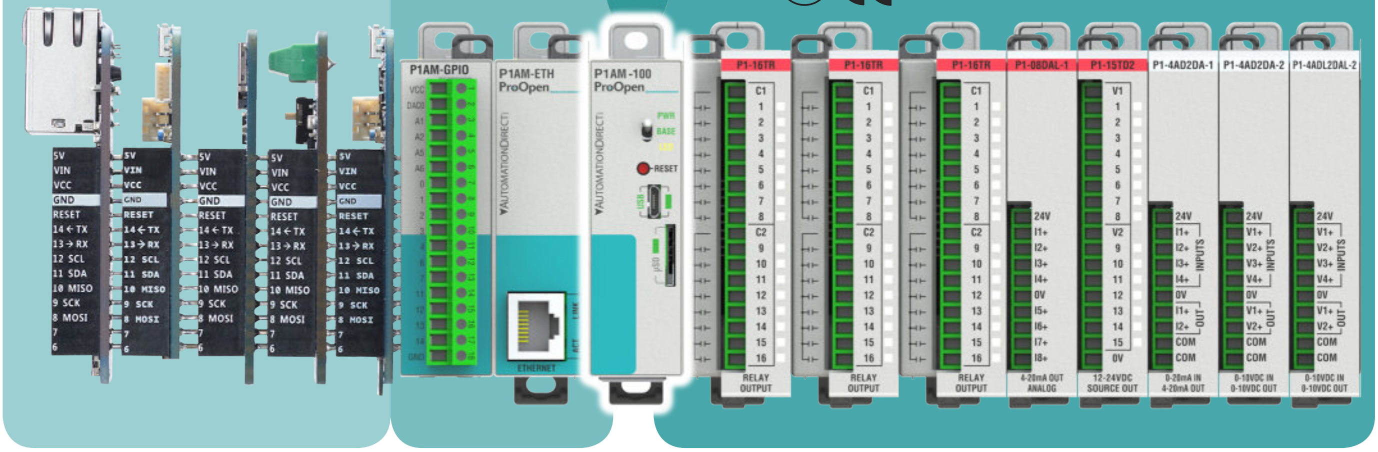
Productivity® Blocks

Open-source MKR shields

Generic and Arduino-brand shields including DIY modules

AutomationDirect industrial I/O, shields, and power supplies

- Power Supplies
- Industrial-grade Shields
- Discrete I/O Modules
- Analog/Temperature I/O Modules
- Relay I/O Modules
- Specialty I/O Modules



P1AM-100 CPU

MAKER IN...INDUSTRIAL OUT

Reducing the risk of open source

With the growing popularity of single-board controllers and the risks involved with implementing them in industrial applications, it was apparent that our industry needed an open-source controller that would hold up in the most extreme conditions. Produced in conjunction with FACTS Engineering, the ProductivityOpen controller (P1AM-100) is just that as it combines the best of both worlds - Maker ingenuity coupled with our Productivity controller family's proven reliability.

Productivity® Open

With the ProductivityOpen platform you get all the great features of a standard Arduino plus the added power and reliability of an industrial controller.

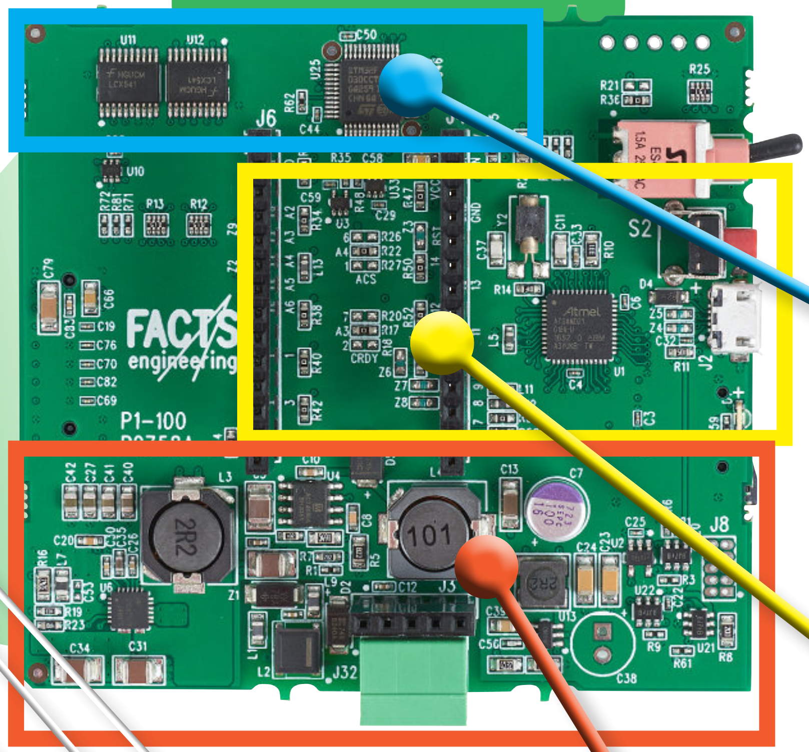
- Open-source sample code available
- C++ programming
- Add-on shields for product optimization
- Low cost controller

+

- Industrialized power supply for EMI and ESD protection
  - UL61010 listed for industrial applications
  - Expandable I/O with over 240 points using right-side I/O modules and left-side shields
- Compatible with P1000 I/O modules (also UL61010)
- Industrial-grade shields for Ethernet, GPIO and prototyping
- ProductivityBlocks, intuitive block programming software
- 2 year warranty

Industrial-Grade Arduino-Compatible CPU

WHAT'S INSIDE:

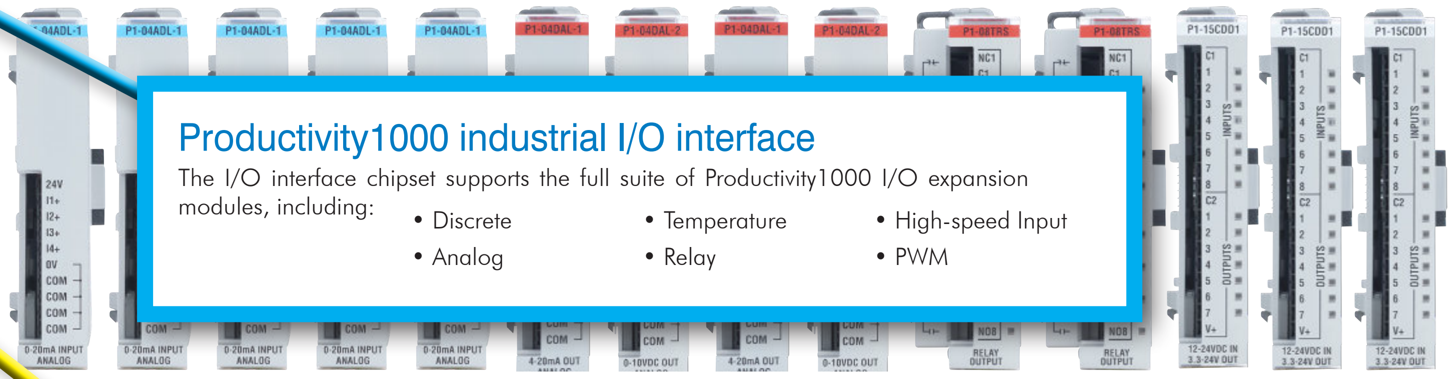


Arduino Onboard LED (under user control)

Run/Stop Switch (under user control)

USB Interface

SD Card Slot



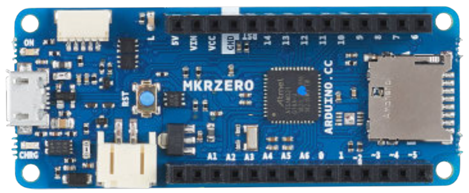
Productivity1000 industrial I/O interface

The I/O interface chipset supports the full suite of Productivity1000 I/O expansion modules, including:

- Discrete
- Analog
- Temperature
- Relay
- High-speed Input
- PWM

Arduino inside

The processor circuit of the P1AM-100 is designed to mimic the Arduino MKRZero microcontroller. By doing this, the P1AM-100 is able to recognize most available Arduino MKR format shields, and/or all of the industrially-hardened Productivity shields, and can utilize most Arduino sketch programs found on open-source websites. Using the same IDE, the P1AM-100 is programmed using C++ code, or you can use the ProductivityBlocks graphical programming interface to quickly code the controller to operate your application.



Industrial power supply stage

The robust power supply filtering stage produces a regulated 5VDC output from a 24VDC input, isolating the CPU and I/O power. To generate the 24VDC input, use any of the field-proven Productivity1000 industrial power supplies or supply your own using the terminal block connection.

Things to consider when choosing between a PLC and Maker controller

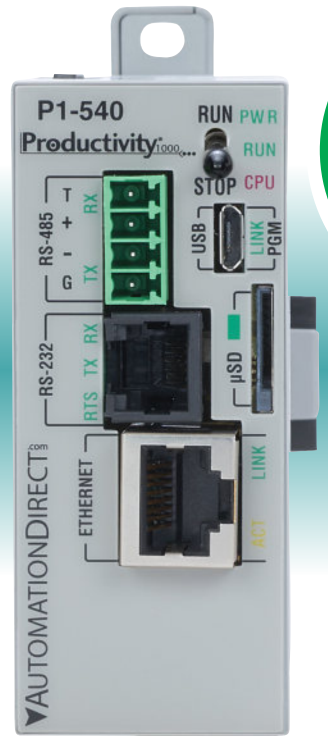
For the PLC'ers:

Let's be honest, a CPU is definitely something worthy of a closer look. But for those coming from a strictly PLC background there are some things to be aware of. Besides the obvious difference of programming methods (C++ vs. Ladder Logic) there are some other functional differences that also need to be addressed and we've included them in the table below.

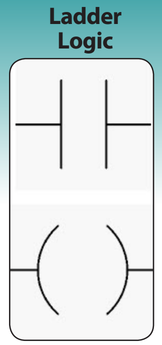
Industrial Controller Comparison	**P1AM-100** **Arduino-compatible CPU**	**P1-540** **PLC CPU**
Programming language	C/C++, ProductivityBlocks, Other Community	Ladder Logic
Development environment	Arduino IDE, Other Community; blank slate no native functions like PID	Productivity Suite; built-in instructions like PID, communication drivers and support functions
Form factor	P1000	P1000
Right-side expansion (I/O modules)	P1000	P1000
Left-side expansion (shields)	P1AM Family, Arduino MKR form factor shields	N/A
Power supply	P1 family or AUX-in without 3rd party shields; AUX-in with 3rd party shields	P1 family, AUX-in
Interfaces	USB Programming, Arduino MKR expansion bus	USB Programming, RS232/485, Ethernet
CPU toggle switch	User controlled	Run/Stop system controlled
User LED	User controlled	System controlled
Memory: project memory	256kB flash	50,000kB
Memory: data retentive	N/A	500kB
Memory: removable media	µSD	µSD
Custom electronic designs	P1AM-PROTO	N/A
3rd party expansion	Yes, using Arduino MKR expansion bus	N/A
Project stored on CPU	No, only binary executable file is stored on CPU; executable file cannot be retrieved from CPU	Yes, optionally
I/O update control	Typically immediately within program instructions	Typically at beginning/end of scan loop
GUI FW updates	Controlled by Arduino.cc	Ugraded by user
Board and library updates	Auto update based on user settings	Manual SW/FW updates from AutomationDirect.com
IDE updates	Arduino IDE from Arduino.cc and others	Productivity Suite Software from AutomationDirect.com
Community sharing	Open source, community driven sharing of programs and support	N/A
Online/runtime edits	N/A	Yes
Auto-configured I/O	N/A	Yes

Industrial-Grade
Arduino-Compatible CPU

Productivity Open



Industrial-Grade
PLC CPU



Productivity 1000

For the Makers:

Those of you who are very familiar with open-source controllers, like the Arduino, may be wondering what an industrial controller could provide. Besides the ruggedness and survivability, there are many other benefits as well, some of which are covered in the table below.

Arduino/Industrial Controller Comparison	**Arduino**	**P1AM-100** **(Arduino-compatible CPU)**	**P1-540** **(PLC CPU)**
Analog I/O	0-3.3V I/O PWM outputs	Wide variety of analog I/O options available including 0-20 mA, 0-10V, PWM and RTD	
Wiring options	Screw type terminal block	Screw type or spring clamp terminal blocks and prewired connections (ZipLinks)	
DINrail mounting	N/A	Yes	
External watchdog functionality	N/A	Yes	
Resources	Multiple; hobby focus	Multiple; industrial automation focus	
Programming instruction	N/A		Yes
Development timesavers	ArduBlock user interface	ProductivityBlocks user interface	Built-in instructions, tag import/export, I/O auto-discovery, etc.

Proven hardware that won't let you down

(P1-01AC)

Power Supplies

Productivity1000 power supplies provide 16 or 26 W of output power with VDC or VAC input options.

- P1-01DC - 12-24 VDC input with 24VDC, 0.67 A, 16W output.
- P1-01AC - 100-240 VAC or 125VDC input with 24VDC, 0.67 A, 16W output.
- P1-02AC - 100-240 VAC or 125VDC input with 24VDC, 1.08 A, 26W output.

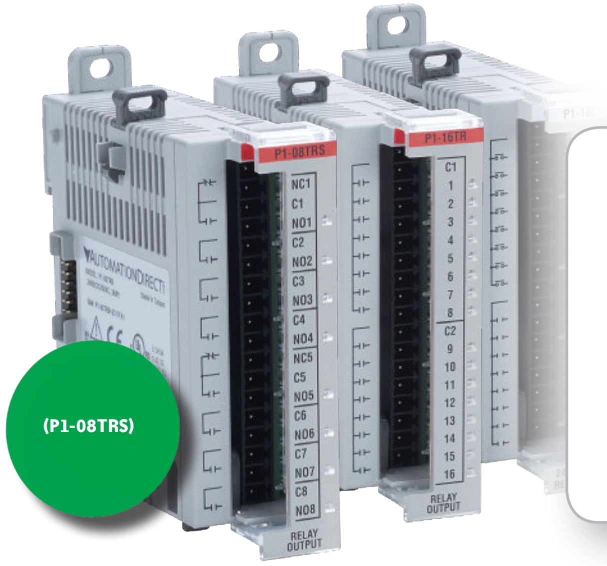
NOTE: You can use your own 24VDC power supply by wiring directly to the P1AM-100 CPU power terminals

(P1-08TD2)

Discrete I/O Modules

Discrete input, output and combo input/output modules are available in 8 or 16-point versions with various DC/AC voltage ranges.

- P1-08ND3 - Input Module 8-pt, 12-24 VDC
- P1-08NE3 - Input Module 8-pt, 24 VAC/VDC
- P1-08NA - Input Module: 8-pt, 120-240 VAC
- P1-16ND3 - Input Module: 16-pt, 12-24 VDC
- P1-16NE3 - Input Module: 16-pt, 24 VAC/VDC
- P1-08TD1 - Output Module 8-pt, 3.3-24 VDC
- P1-08TD2 - Output Module 8-pt, 12-24 VDC
- P1-08TA - Output Module: 8-pt, 120-240 VAC
- P1-15TD1 - Output Module: 15-pt, 3.3-24 VDC
- P1-15TD2 - Output Module: 15-pt, 12-24 VDC
- P1-15CDD1 - Combo Module 8-pt 12-24 VDC in, 7-pt 3.3-24 VDC out
- P1-15CDD2 - Combo Module 8-pt 12-24 VDC in, 7-pt 12-24 VDC out

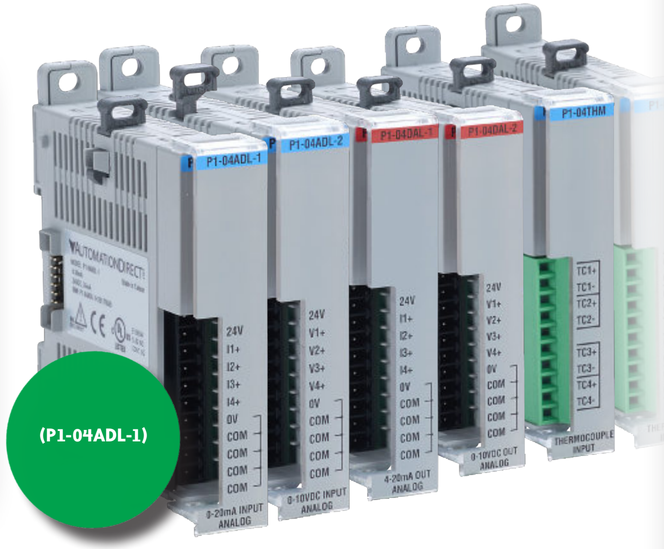


(P1-08TRS)

Relay I/O Modules

Relay output modules support devices that operate with voltages up to 240VAC or 24VDC.

- P1-08TRS - Output Module 8-pt, 6-24 VDC/6-120 VAC, 3A/pt
- P1-16TR - Output Module 16-pt, 6-24 VDC/6-240 VAC, 2A/pt
- P1-16CDR - Combo Module 8-pt discrete 24 VAC/VDC in, 8-pt 6-24 VDC/6-240 VAC relay out, 1A/pt



(P1-04ADL-1)

Analog/Temperature I/O Modules

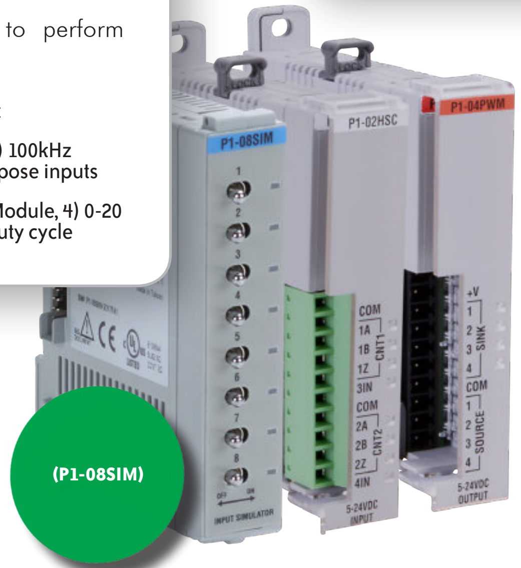
Analog input and output modules are available to monitor and control pressure, temperature, flow, level or any other process signal your application requires.

- P1-04AD - Input Module 4-channel, ±5VDC, ±10VDC, 0-5 VDC, 0-10 VDC and 0-20 mA, 16-bit resolution
- P1-04ADL-1 - Input Module 4-channel, 0-20 mA, 13-bit resolution
- P1-04ADL-2 - Input Module 4-channel, 0-10 VDC, 13-bit resolution
- P1-08ADL-1 - Input Module 8-channel, 0-20 mA, 13-bit resolution
- P1-08ADL-2 - Input Module 8-channel, 0-10 VDC, 13-bit resolution
- P1-04DAL-1 - Output Module 4-channel, 4-20 mA, 12-bit resolution
- P1-04DAL-2 - Output Module 4-channel, 0-10 VDC, 12-bit resolution
- P1-4ADL2DAL-1 - Combo Module 4-channel 0-20 mA in, 2-channel 4-20 mA out
- P1-4ADL2DAL-2 - Combo Module 4-channel 0-10 VDC in, 2-channel 0-10 VDC out
- P1-08DAL-1 - Output Module 8-channel, 4-20 mA, 12-bit resolution
- P1-08DAL-2 - Output Module 8-channel, 0-10 VDC, 12-bit resolution
- P1-04THM - Thermocouple Input Module 4-channel, 16-bit resolution
- P1-04NTC - Thermistor Input Module 4-channel, 16-bit resolution
- P1-04RTD - RTD Input Module, 4-channel, 16-bit resolution

Specialty I/O Modules

Specialty modules are designed to perform specific functions.

- P1-08SIM - Input Simulator Module, 8-pt
- P1-02HSC - High-speed Input Module, 2) 100kHz counter inputs, 2) 5-24 VDC general purpose inputs
- P1-04PWM - Pulse Modulation Output Module, 4) 0-20 kHz pulse modulated outputs, 0-100% duty cycle

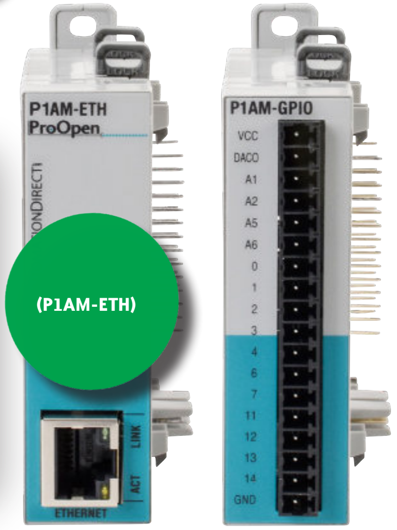


(P1-08SIM)

Industrial-grade Shields

ProductivityOpen industrial shields are rated for harsh duty and can add needed functionality to the P1AM-100 CPU.

- P1AM-ETH - Ethernet Module, provides a single 10/100 Mbps Ethernet connection
- P1AM-GPIO - General Purpose I/O Module, subset of MKR header pins routed to front-facing 18-pt terminal block



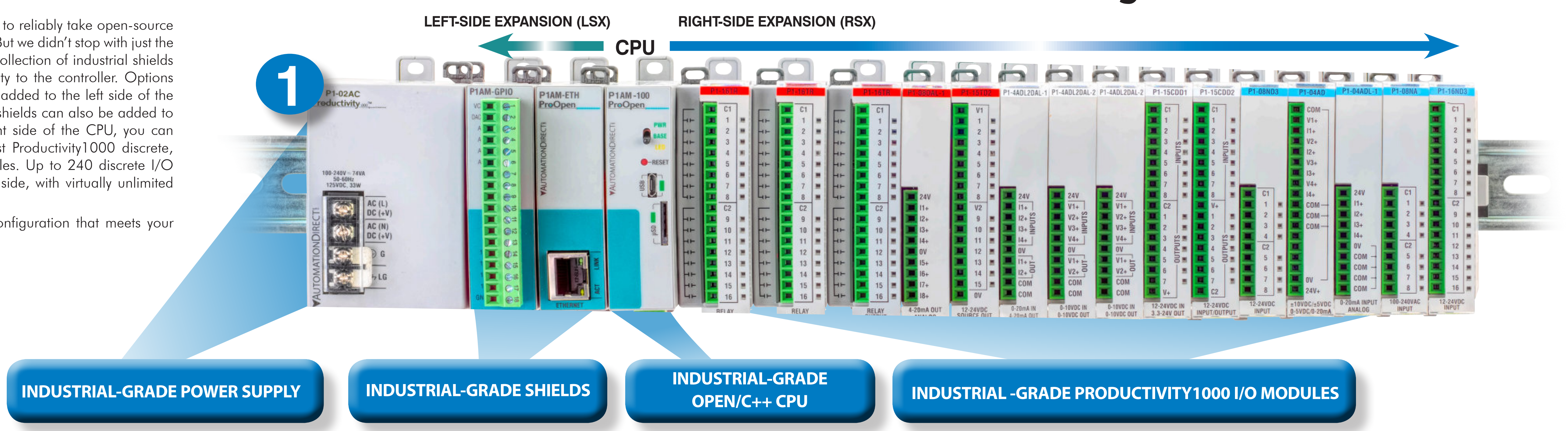
(P1AM-ETH)

4 expansion options for maximum versatility

Productivity® Open

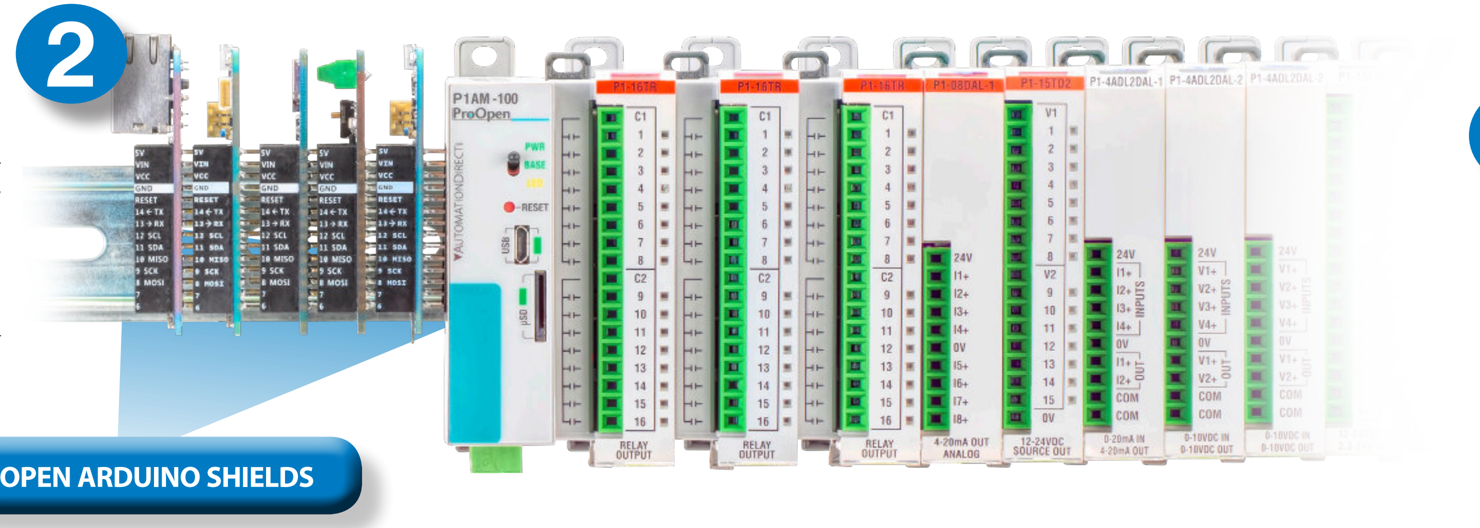
The P1AM-100 CPU is designed to reliably take open-source control into the industrial realm. But we didn't stop with just the CPU. We've also engineered a collection of industrial shields that can add needed functionality to the controller. Options including Ethernet can easily be added to the left side of the CPU. Readily-available Arduino shields can also be added to that side if needed. On the right side of the CPU, you can expand the system with low-cost Productivity1000 discrete, analog and specialty I/O modules. Up to 240 discrete I/O points are possible on the right-side, with virtually unlimited I/O on the left.

It's your choice to select any configuration that meets your needs:


LEFT-SIDE EXPANSION (LSX)
CPU
RIGHT-SIDE EXPANSION (RSX)
1
P1-02AC
P1AM-GPIO
P1AM-ETH
P1AM-100
INDUSTRIAL-GRADE POWER SUPPLY
INDUSTRIAL-GRADE SHIELDS
INDUSTRIAL-GRADE OPEN/C++ CPU
INDUSTRIAL -GRADE PRODUCTIVITY1000 I/O MODULES

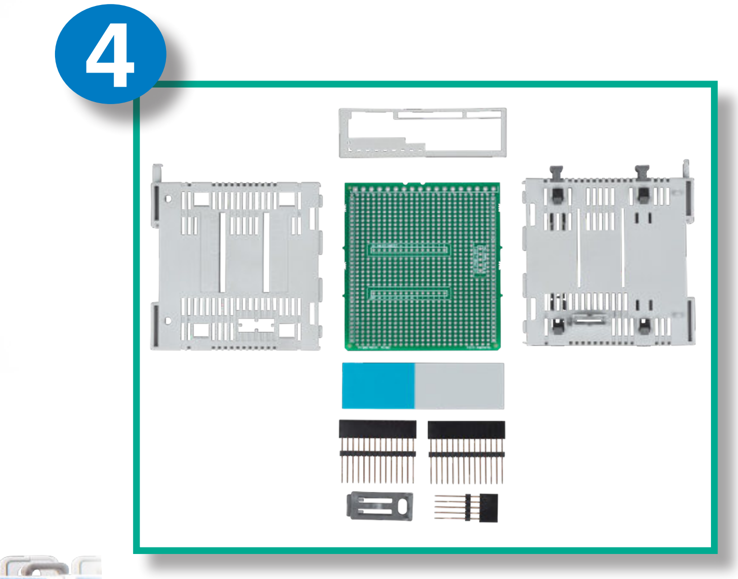
1. 100% industrial

Ensure that all aspects of your open-source controller are protected from harsh environments with industrially-rated power supplies, shields, CPU and I/O modules.



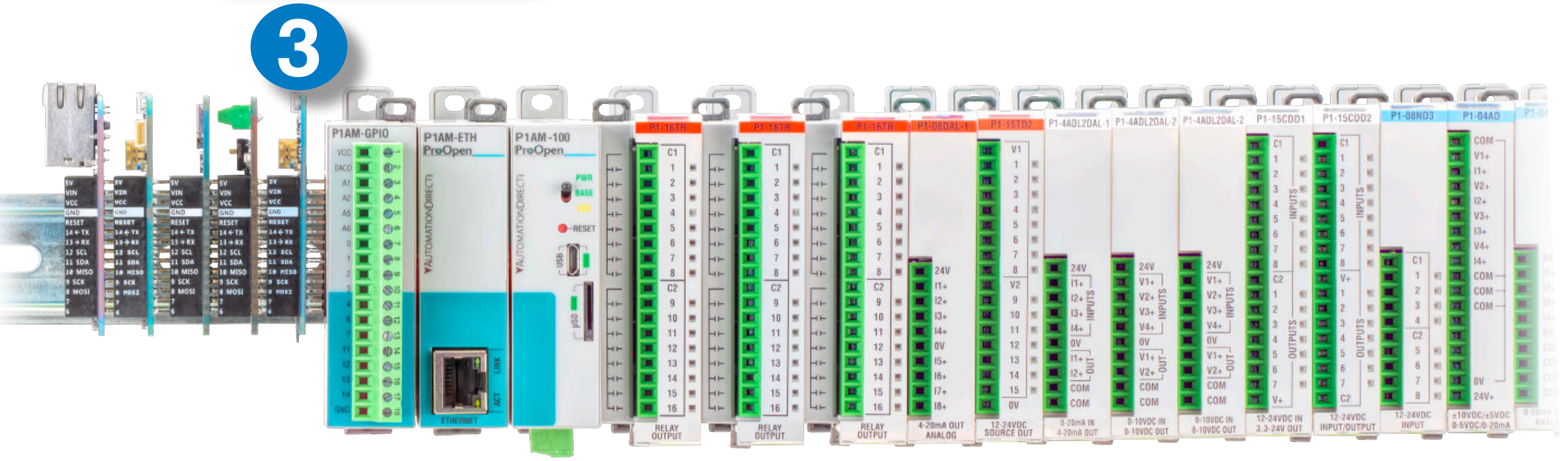
2. Industrial Maker

Got a specific Arduino shield you want to use in your process? That's perfectly okay with the ProductivityOpen controller. Simply attach any compatible Arduino shield(s)* to the left side of the CPU and use Productivity1000 industrially-hardened I/O modules to give your controller added protection from field equipment.



4. DIY all the way

Build custom electronic circuits and interfaces for your control system with our proto board. The P1AM-PROTO is a generic perf board with 100mil thru-holes for your own prototype designs.



3. The jack of all trades

You can mix and match any combination of compatible open Arduino shields* and industrially-rated ProductivityOpen shields to achieve the control you're looking for. The ProductivityOpen controller has been designed to provide the utmost flexibility to satisfy both Makers and industry professionals.

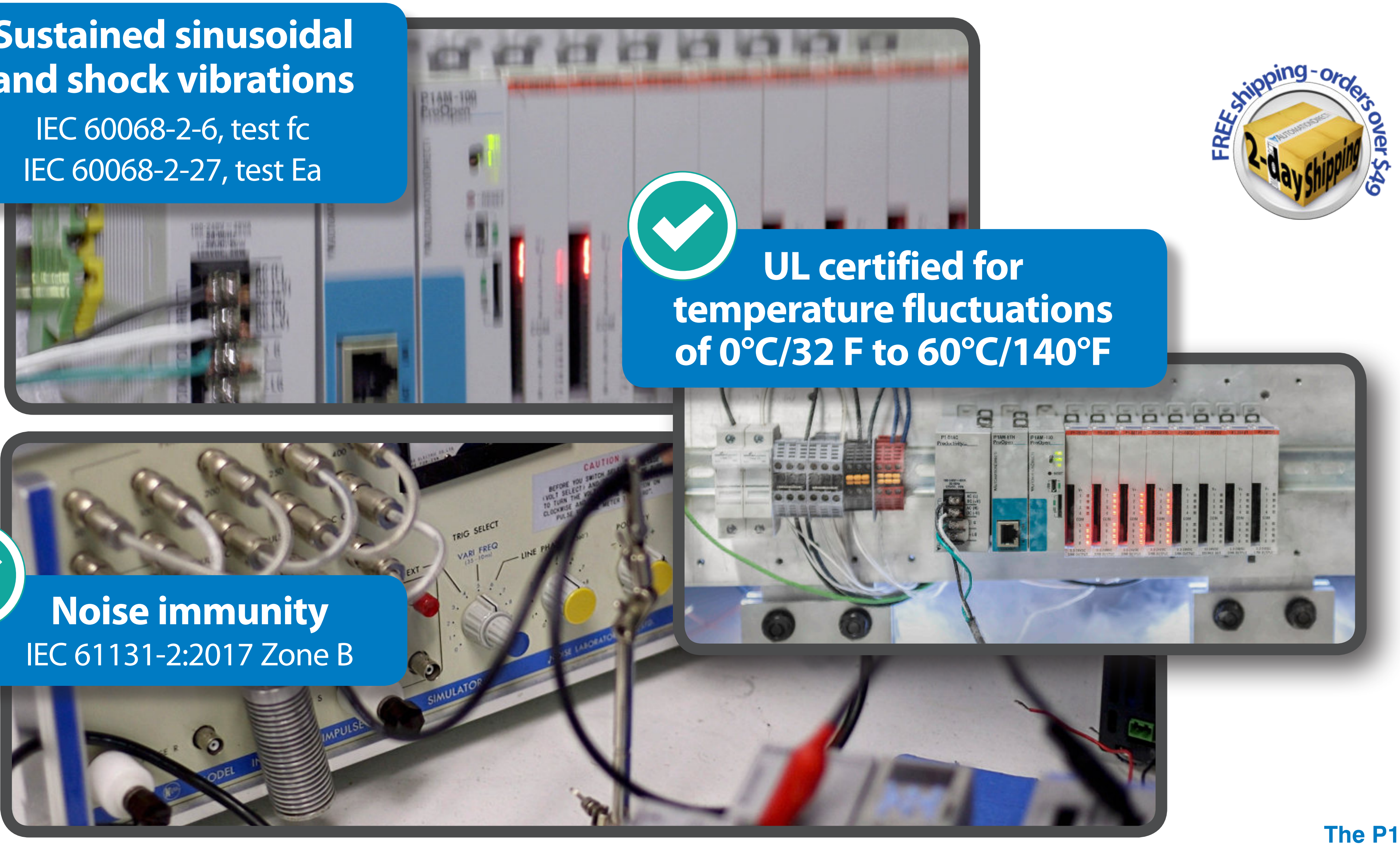
Use discretion, since many of the consumer-grade Arduino shields are not suitable for industrial applications.

Tested, tested, and tested again to ensure quality

Why should UL have all the fun?

FACTS Engineering, our development and manufacturing partner for Productivity controllers, takes product reliability very seriously. When developing new control products like the P1AM-100, FACTS thoroughly tests them in house to validate their longevity. Once the product has been through FACTS' rigorous testing, there's really no doubt they'll be certified by UL.

FACTS has many in-house testing stations at their facility in New Port Richey, FL, including a shake table and temperature chamber that they use to ensure your controller continues to perform, no matter how harsh the environment, well beyond the purchase.



Sustained sinusoidal and shock vibrations

IEC 60068-2-6, test fc

IEC 60068-2-27, test Ea

UL certified for temperature fluctuations of 0°C/32 F to 60°C/140°F

Noise immunity

IEC 61131-2:2017 Zone B

Getting started is easy with our convenient starter kits!

Our starter kits provide everything needed to get you on your way. CPU, industrial shields, industrial I/O modules, power cables and more are all included with the P1AM-START1 kit. This kit is intended for industrial Makers who are ready to get a jump on their next project. The P1AM-START2 is a lower-cost starter kit without industrial shields and includes CPU, industrial I/O, power supply, etc., perfect for those wanting to learn more about open-source control.

P1AM-START1

What's in the P1AM-START1? A kit for the devoted Maker:

P1AM-START1

- (1) P1AM-100 CPU
- (1) P1AM-ETH Ethernet shield
- (1) P1AM-GPIO general purpose I/O shield
- (1) P1-4ADL2DAL-1 analog combo module
- (1) PSL-24-030 power supply
- (1) USB-CBL-AMICB6 programming cable
- (1) 3-wire power cable
- (1) P2-RTB terminal block
- (1) P1-10RTB terminal block

What's in the P1AM-START2? A kit for the PLC'er wanting to learn Arduino:

P1AM-START2

- (1) P1AM-100 CPU
- (1) P1-08TRS relay output module
- (1) P1-08SIM simulator input module
- (1) P1-01AC power supply
- (1) USB-CBL-AMICB6 programming cable
- (1) 3-wire power cable
- (1) P2-RTB terminal block

The P1AM-100 open-source controller is designed to survive where others fail and we guarantee it with a two-year warranty!

Exceeding the needs of an ever-changing industry

As industry changes, we are there for you!

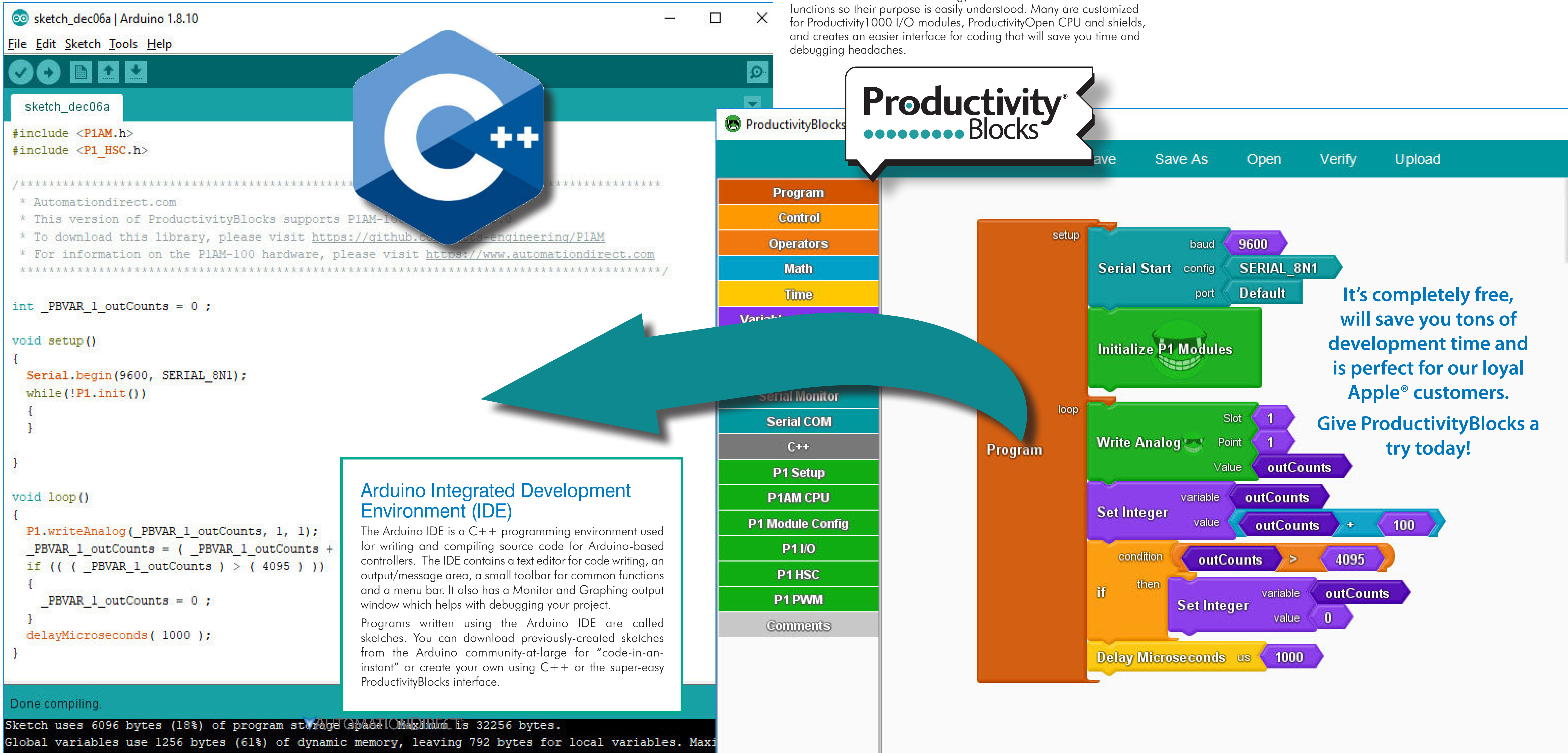
Let's face it, with technology, change is constant. New advancements and techniques are always on the horizon and one major shift we see today is in controller programming. Ladder Logic is still a very popular programming method but other methods, like C++ programming, are making big inroads into industrial automation thanks to low-cost microcontrollers like the Arduino. But keeping up with industry trends doesn't mean you have to sacrifice system integrity.

The P1AM-100 provides the C++ programming environment you want with the industrially-hardened hardware you need.

ProductivityBlocks

Based on the ArduBlock concept, ProductivityBlocks is a graphical programming interface and add-on to the Arduino IDE. If you have ever programmed with C++, you know how tedious it can be hunting down the dreaded syntax error like a missing semicolon or bracket. ProductivityBlocks helps you build your sketch program by dragging and dropping interlocking blocks; the associated C++ is generated for you!

ProductivityBlocks works with either MAC or PC systems offering custom blocks that use terminology common to industrial controller functions so their purpose is easily understood. Many are customized for Productivity1000 I/O modules, ProductivityOpen CPU and shields, and creates an easier interface for coding that will save you time and debugging headaches.



```
#include <P1AM.h>
#include <P1_HSC.h>

/***********************************************            ***************
 * Automationdirect.com
 * This version of ProductivityBlocks supports P1AM-10        .0
 * To download this library, please visit https://github.co      s-engineering/P1AM
 * For information on the P1AM-100 hardware, please visit https://www.automationdirect.com
 ****************************************************************************/

int _PBVAR_1_outCounts = 0 ;

void setup()
{
  Serial.begin(9600, SERIAL_8N1);
  while(!P1.init())
  {
  }

}

void loop()
{
  P1.writeAnalog(_PBVAR_1_outCounts, 1, 1);
  _PBVAR_1_outCounts = ( _PBVAR_1_outCounts +
  if (( ( _PBVAR_1_outCounts ) > ( 4095 ) ))
  {
    _PBVAR_1_outCounts = 0 ;
  }
  delayMicroseconds( 1000 );
}
```

Done compiling.

```
Sketch uses 6096 bytes (18%) of program storage space. Maximum is 32256 bytes.
Global variables use 1256 bytes (61%) of dynamic memory, leaving 792 bytes for local variables. Maxi
```

Arduino Integrated Development Environment (IDE)

The Arduino IDE is a C++ programming environment used for writing and compiling source code for Arduino-based controllers. The IDE contains a text editor for code writing, an output/message area, a small toolbar for common functions and a menu bar. It also has a Monitor and Graphing output window which helps with debugging your project.

Programs written using the Arduino IDE are called sketches. You can download previously-created sketches from the Arduino community-at-large for "code-in-an-instant" or create your own using C++ or the super-easy ProductivityBlocks interface.

It's completely free, will save you tons of development time and is perfect for our loyal Apple® customers.

Give ProductivityBlocks a try today!

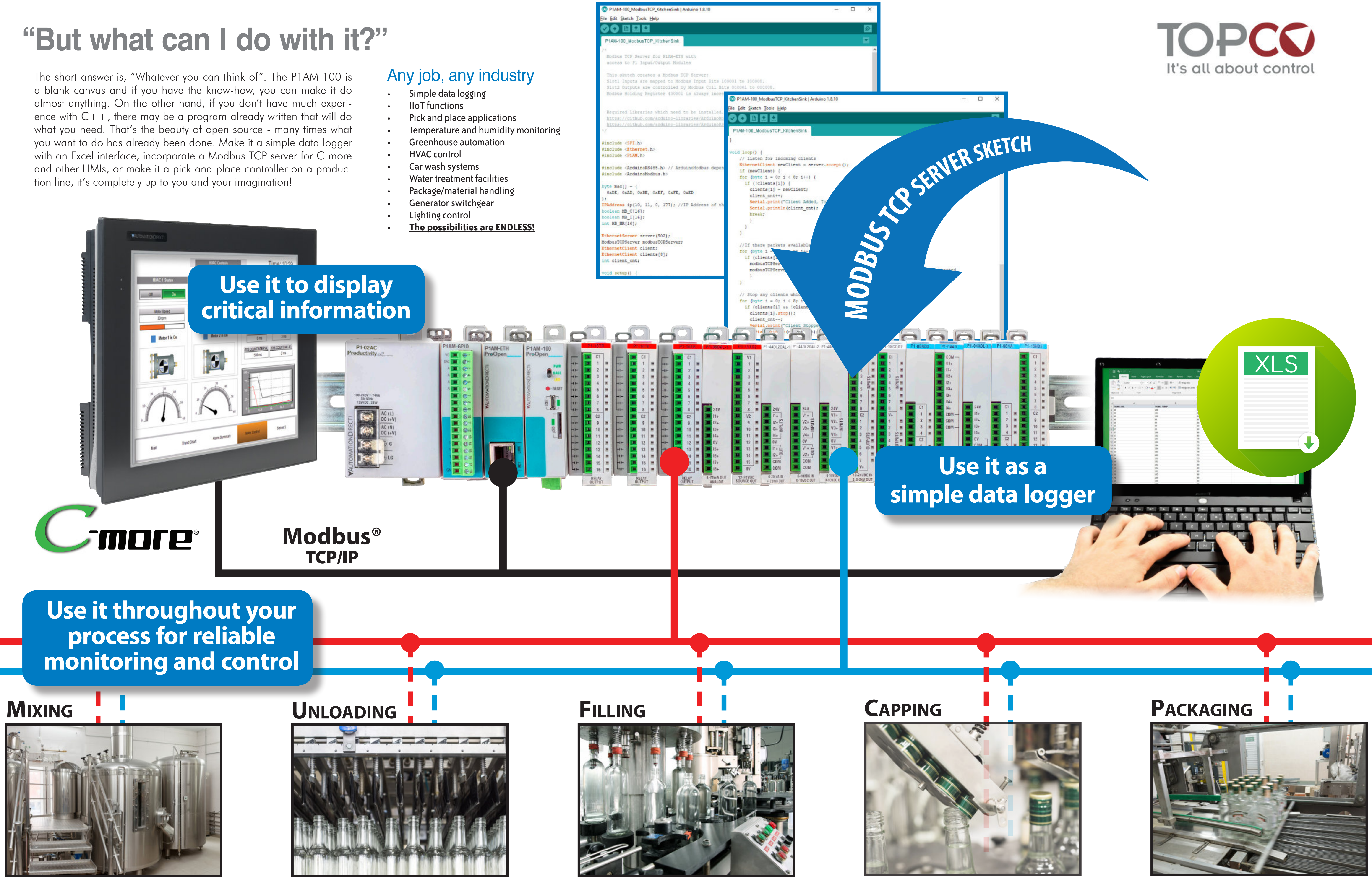
“But what can I do with it?”

The short answer is, “Whatever you can think of”. The P1AM-100 is a blank canvas and if you have the know-how, you can make it do almost anything. On the other hand, if you don’t have much experience with C++, there may be a program already written that will do what you need. That’s the beauty of open source - many times what you want to do has already been done. Make it a simple data logger with an Excel interface, incorporate a Modbus TCP server for C-more and other HMIs, or make it a pick-and-place controller on a production line, it’s completely up to you and your imagination!

Any job, any industry

- Simple data logging
- IIoT functions
- Pick and place applications
- Temperature and humidity monitoring
- Greenhouse automation
- HVAC control
- Car wash systems
- Water treatment facilities
- Package/material handling
- Generator switchgear
- Lighting control
- **The possibilities are ENDLESS!**

TOPCO
It's all about control



P1AM-100_ModbusTCP_KitchenSink | Arduino 1.8.10

```
/*
  Modbus TCP Server for P1AM-ETH with
  access to P1 Input/Output Modules

  This sketch creates a Modbus TCP Server:
  Slot1 Inputs are mapped to Modbus Input Bits 100001 to 100008.
  Slot2 Outputs are controlled by Modbus Coil Bits 000001 to 000008.
  Modbus Holding Register 400001 is always incre


  Required Libraries which need to be installed.
  https://github.com/arduino-libraries/ArduinoMo
  https://github.com/arduino-libraries/ArduinoRS
*/

#include <SPI.h>
#include <Ethernet.h>
#include <P1AM.h>

#include <ArduinoRS485.h> // ArduinoModbus depen
#include <ArduinoModbus.h>

byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(10, 11, 0, 177); //IP Address of th
boolean MB_C[16];
boolean MB_I[16];
int MB_HR[16];

EthernetServer server(502);
ModbusTCPServer modbusTCPServer;
EthernetClient client;
EthernetClient clients[8];
int client_cnt;

void setup() {
```

P1AM-100_ModbusTCP_KitchenSink | Arduino 1.8.10

```
}

void loop() {
  // listen for incoming clients
  EthernetClient newClient = server.accept();
  if (newClient) {
  for (byte i = 0; i < 8; i++) {
    if (!clients[i]) {
      clients[i] = newClient;
      client_cnt++;
      Serial.print("Client Added, To
      Serial.println(client_cnt);
      break;
      }
    }
  }

  //If there packets available
  for (byte i =
    if (clients[i
      modbusTCPSer
      modbusTCPServe
      }
  }

  // Stop any clients whic
  for (byte i = 0; i < 8; i
    if (clients[i] && !client
      clients[i].stop();
      client_cnt--;
      Serial.print("Client Stoppe
```