Open-Source Agility Meets Industrial-Grade Toughness

P1AM-100 CPU
(Arduino-compatible)
$69.00 C++ UL-certified 48MHz CPU

P1AM-200 CPU
(CircuitPython)
$109.00 C++/CircuitPython UL-certified 120MHz CPU

Industrial-Grade Open-Source CPUs starting at $69.00

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 CPUs
Rugged, open-source CPUs compatible with the MKRZero Arduino seamlessly bridge standard 3rd-party shields with AutomationDirect industrial PLC I/O.
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. The ProductivityOpen controllers, produced in conjunction with FACTS Engineering, provide an ideal solution as they combine the best of both worlds - Maker ingenuity coupled with our Productivity controller family’s proven reliability.

Industrial-Grade Open-Source CPUs starting at $69.00

WHAT’S INSIDE:

- Onboard LED (under user control)
- Run/Stop Switch (under user control)
- USB Interface
- SD Card Slot

Overview Video: See what ProductivityOpen has to offer!

Click above or go to http://go2adc.com/p1am-overview to view

With the ProductivityOpen controllers you get all the great benefits of an open-source platform plus the added power and reliability of an industrial controller.

Open source at heart
The processor circuit of the P1AM-CPU is designed to mimic the Arduino MKRZero microcontroller. By doing this, the CPU is able to recognize most available Arduino MKR format shields and/or all of the industrially-hardened Productivity shields. The P1AM-100 CPU uses the Arduino IDE and can be programmed using C++ code or it can utilize most Arduino sketch programs found on open-source websites. The P1AM-200 CPU utilizes CircuitPython programming and can be coded using any text editor or IDE, or C++ using the Arduino IDE. Numerous sample programs are also available for CircuitPython that can be easily downloaded to the P1AM-200 CPU.

*The P1AM-200 CPU is also equipped with a Crypto coprocessor ensuring performance doesn’t suffer when using encryption libraries.

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

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

*neopixel LED also included on P1AM-200 CPU

For the latest prices, please check AutomationDirect.com.
Open source the Productivity way!

What is “open source”?
The term “open source” is used to describe a program or software created by one developer that is available to be used and/or modified in any way by other developers and users without licensing fees, royalties due, or restrictions on the use of the source code. This is sometimes referred to as “copyleft” as opposed to “copyright”. Open source has evolved to also include hardware, shared schematics and PCB production files that are often readily available to anyone. This type of shared development has spawned an enormous “Maker” community. Numerous Maker sites can be found online with a vast collection of simple, helpful and mostly all reusable DIY projects.

The microcontrollers used to run these DIY programs are inexpensive, small and typically consist of a single integrated circuit containing a processor, memory and I/O. A brand of single-board microcontrollers that has become one of the most well-known is the Arduino.

What is Arduino?
Arduino products were originally created for students without backgrounds in electronics or computer programming. Arduino consists of a family of single programmable circuit boards and the IDE (Integrated Development Environment) that uses a streamlined version of C++ to write and upload code to the boards. Many pre-configured circuit boards, called “shields”, are available to expand the functionality of the Arduino controller. These shields can provide Ethernet, WiFi, GPS, LCD displays, and motor controls, among others, by simply “stacking” or connecting the shields to the Arduino controller board.

Avoid the ticking time bomb
The open-source concept is favored heavily by hobbyists and students, but recently the industrial controls industry has also taken notice, partly due to the extremely attractive price tag. Industrial applications using “off-the-shelf” Arduinos have increased, but there is a risk with installing these single-board controllers in industrial environments. Many of these controllers are not field tested and in most instances are just downtime waiting to happen.

Vibration, noise, and temperature fluctuations can have a negative effect on consumer-grade microcontrollers, causing unexpected equipment failures and costly production shutdowns. In these types of harsh applications, you need a controller designed to survive - you need the ProductivityOpen!

Open-source communities
Sharing of ideas and finding innovative ways to solve complex problems is facilitated by open-source communities and the websites dedicated to them. Sites like MakerPro and GitHub allow hobbyists and professionals to work together to create interesting solutions for difficult or everyday problems.

“Open-source communities
Sharing of ideas and finding innovative ways to solve complex problems is facilitated by open-source communities and the websites dedicated to them. Sites like MakerPro and GitHub allow hobbyists and professionals to work together to create interesting solutions for difficult or everyday problems.

“Open-source communities
Sharing of ideas and finding innovative ways to solve complex problems is facilitated by open-source communities and the websites dedicated to them. Sites like MakerPro and GitHub allow hobbyists and professionals to work together to create interesting solutions for difficult or everyday problems.

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

Open product Open

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

Open product Open

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

Open product Open

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

Open product Open

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

Open product Open

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”

“A simple DIY automatic pet feeder I made using a single board controller I got from Amazon.”

“I could use the pet feeder code to create an automatic pig feeder on my farm... ...But I need a controller that can survive the extremely hot Texas summers.”

“I could alter the feeder code to automatically feed ingredients into a mixing tank... ...But I need a controller that can handle the plant’s harsh environment.”
Things to consider when choosing between a PLC and Maker controller

For the PLC’ers:
Let’s be honest, a $69.00 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 there are some other functional differences that also need to be addressed and we’ve included them in the table below.

### Industrial Controller Comparison

<table>
<thead>
<tr>
<th>P1AM-100 (Microcontroller CPU)</th>
<th>P1AM-200 (Microcontroller CPU)</th>
<th>P1-540 (PLC CPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming language</strong></td>
<td>C/C++ or Arduino legacy IDE</td>
<td>Ladder logic</td>
</tr>
<tr>
<td></td>
<td>Other community</td>
<td></td>
</tr>
<tr>
<td><strong>Development environment</strong></td>
<td>Arduino IDE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other Community</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blank state no native functions like PID</td>
<td></td>
</tr>
<tr>
<td><strong>Form factor</strong></td>
<td>Productivity 1000</td>
<td></td>
</tr>
<tr>
<td><strong>Right-side expansion</strong></td>
<td>Productivity 1000</td>
<td></td>
</tr>
<tr>
<td>(I/O modules)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Left-side expansion</strong></td>
<td>P1AM family</td>
<td>N/A</td>
</tr>
<tr>
<td>(shields)</td>
<td>Arduino MKR form factor shields</td>
<td></td>
</tr>
<tr>
<td><strong>Interfaces</strong></td>
<td>USB programming</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arduino MKR expansion bus</td>
<td></td>
</tr>
<tr>
<td><strong>CPU toggle switch</strong></td>
<td>User controlled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run/Stop system controlled</td>
<td></td>
</tr>
<tr>
<td><strong>User LED</strong></td>
<td>User controlled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System controlled</td>
<td></td>
</tr>
<tr>
<td><strong>Memory: project memory</strong></td>
<td>256kB flash</td>
<td>50MB</td>
</tr>
<tr>
<td></td>
<td>16MB flash</td>
<td></td>
</tr>
<tr>
<td><strong>Memory: data retentive</strong></td>
<td>N/A</td>
<td>500kB</td>
</tr>
<tr>
<td><strong>Memory: removable media</strong></td>
<td>µSD</td>
<td></td>
</tr>
<tr>
<td><strong>3rd party expansion</strong></td>
<td>Yes, using Arduino MKR</td>
<td></td>
</tr>
<tr>
<td>expansion bus</td>
<td>expansion bus</td>
<td></td>
</tr>
<tr>
<td><strong>Project stored on CPU</strong></td>
<td>No, only binary executable file</td>
<td>Yes, optionally</td>
</tr>
<tr>
<td></td>
<td>stored on CPU; executable file cannot be retrieved from CPU</td>
<td></td>
</tr>
<tr>
<td><strong>I/O update control</strong></td>
<td>Typically immediately within</td>
<td>Typically at</td>
</tr>
<tr>
<td></td>
<td>program instructions</td>
<td>beginning/end of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scan loop</td>
</tr>
<tr>
<td><strong>GUI/FW updates</strong></td>
<td>Controlled by Arduino.cc</td>
<td>Ugraded by user</td>
</tr>
<tr>
<td><strong>Board and library updates</strong></td>
<td>Auto update based on user</td>
<td>Manual SW/FW</td>
</tr>
<tr>
<td></td>
<td>settings</td>
<td>updates from</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AutomationDirect.com</td>
</tr>
<tr>
<td><strong>IDE updates</strong></td>
<td>Arduino IDE from Arduino.cc</td>
<td>Productivity Suite Software from AutomationDirect.com</td>
</tr>
<tr>
<td></td>
<td>and others</td>
<td></td>
</tr>
<tr>
<td><strong>Community sharing</strong></td>
<td>Open source; community driven</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sharing of programs and support</td>
<td></td>
</tr>
<tr>
<td><strong>Online/runtime edits</strong></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td><strong>Auto-configured I/O</strong></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td><strong>Processor speed</strong></td>
<td>48MHz</td>
<td>1200MHz</td>
</tr>
<tr>
<td></td>
<td>1.3 ms scan time (CPU clock)+</td>
<td></td>
</tr>
<tr>
<td><strong>Crypto coprocessor</strong></td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>$69.00</td>
<td>$109.00</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Arduino (MKR ZERO)</th>
<th>P1AM-100/200 (Microcontroller CPUs)</th>
<th>P1-540/550 (PLC CPUs)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Power Supply</strong></td>
<td>5VDC</td>
<td>24VDC AUX-in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DC power supply (P1-01DC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC power supplies (P1-01AC, P1-02AC)</td>
</tr>
<tr>
<td><strong>Agency Approvals</strong></td>
<td>CE</td>
<td>UL / CE</td>
</tr>
<tr>
<td><strong>Analog and Digital I/O</strong></td>
<td>3.3VDC tolerant MKR</td>
<td>Productivity 1000 I/O</td>
</tr>
<tr>
<td></td>
<td>Productivity 1000 MKR</td>
<td>Produticy 1000 I/O</td>
</tr>
<tr>
<td><strong>Analog Input Resolution</strong></td>
<td>8,10,12 bit MKR</td>
<td>Productivity 1000 analog inputs</td>
</tr>
<tr>
<td></td>
<td>10 bit MKR</td>
<td>Productivity 1000 analog outputs</td>
</tr>
<tr>
<td><strong>Interrupts</strong></td>
<td>Yes MKR</td>
<td>No</td>
</tr>
<tr>
<td><strong>Serial Communication</strong></td>
<td>MKR UART 3.3VDC tolerant MKR</td>
<td>MKR UART 3.3VDC</td>
</tr>
<tr>
<td></td>
<td>3rd party shields RS323/485</td>
<td>MKR UART 3.3VDC</td>
</tr>
<tr>
<td></td>
<td>Productivity 1000 SERIAL &amp; 3rd</td>
<td>MKR UART 3.3VDC</td>
</tr>
<tr>
<td></td>
<td>party shields RS323/485</td>
<td>MKR UART 3.3VDC</td>
</tr>
<tr>
<td></td>
<td>MKR shield</td>
<td>MKR shield</td>
</tr>
<tr>
<td></td>
<td>MKR-ETH shield</td>
<td>MKR-ETH shield</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td>MKR shield</td>
<td>Productivity 1000 shield</td>
</tr>
<tr>
<td></td>
<td>(1) on P1-540</td>
<td>MKR-ETH shield</td>
</tr>
<tr>
<td></td>
<td>(2) on P1-550</td>
<td>MKR-ETH shield</td>
</tr>
<tr>
<td><strong>I/O Direction Control (GPIO)</strong></td>
<td>Yes MKR</td>
<td>Yes MKR</td>
</tr>
<tr>
<td></td>
<td>MKR-ETH shield</td>
<td>Productivity 1000 shield</td>
</tr>
<tr>
<td><strong>Mounting Options</strong></td>
<td>Breadboard</td>
<td>DIN rail, Screw mount</td>
</tr>
<tr>
<td></td>
<td>Internal</td>
<td>Internal and secondary onboard</td>
</tr>
<tr>
<td><strong>Watchdog</strong></td>
<td>Multi-pinning</td>
<td>Multi-pinning</td>
</tr>
<tr>
<td><strong>IDE Debugging Tools</strong></td>
<td>Serial monitor / plotter</td>
<td>Multi-pinning</td>
</tr>
<tr>
<td></td>
<td>Data view</td>
<td>Multi-pinning</td>
</tr>
<tr>
<td></td>
<td>Monitor view</td>
<td>Multi-pinning</td>
</tr>
<tr>
<td></td>
<td>Debugger</td>
<td>Multi-pinning</td>
</tr>
<tr>
<td></td>
<td>Graphing</td>
<td>Multi-pinning</td>
</tr>
</tbody>
</table>

MKR: Arduino MKR expansion bus
Productivity 1000 I/O: A full line of I/O modules including 12-24 VDC and 3.3-5 VDC inputs, 3.3-24 VDC, 5 VDC, and 6-120 VAC outputs, and relay outputs
Productivity 1000 analog: A full line of A/D, D/A, and temperature input modules, in 12, 13 and 16 bit resolutions

For the latest prices, please check AutomationDirect.com.
Proven hardware that won’t let you down

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 24 VDC, 0.67 A, 16 W output.
- P1-02AC - Input Module, 8-pt, 120-240 VAC, 10 A, 24 W output.
- P1-02AD - Input Module, 8-pt, 120-240 VAC and 125 VDC input with 8 VDC, 0.2 A, 2 W output.

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

Starting at $52.00 (P1-02AC)

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.
- P1-08TD2 - Output Module 8-pt, 12-24 VDC
- P1-08TRS - Output Module 8-pt, 6-24 VDC/6-120 VAC
- P1-08TD1 - Output Module 8-pt, 3.3-24 VDC
- P1-08ADL - Output Module 8-pt, 4-20 mA, 13-bit resolution
- P1-08THL - Output Module 8-pt, 4-20 mA, 16-bit resolution
- P1-08SIM - Input Module 8-pt, 5 VDC
- P1-08SIM2 - Input Module 8-pt, 5 VDC
- P1-08DAL - Output Module 8-pt, 4-20 mA, 12-bit resolution
- P1-04DAL - Output Module 4-channel, 4-20 mA, 12-bit resolution

Starting at $47.50 (P1-08TD2)

Relay I/O Modules
Relay output modules support devices that operate with voltages up to 240 VAC or 24 VDC.
- P1-08TD5 - Output Module 8-pt, 5-24 VDC/4-20 mA
- P1-08TD6 - Upgrade Module 8-pt, 6-24 VDC/6-120 VAC, 3 A/pt

Starting at $61.00 (P1-08TD5)

Relay Output Modules
Relay output modules support devices that operate with voltages up to 240 VAC or 24 VDC.
- P1-08TD5 - Output Module 8-pt, 5-24 VDC/4-20 mA
- P1-08TD6 - Upgrade Module 8-pt, 6-24 VDC/6-120 VAC, 3 A/pt

Starting at $61.00 (P1-08TD5)

Specialty I/O Modules
Specialty modules are designed to perform specific functions.
- P1-08SIM - Input Simulator Module, 8-bit
- P1-06GIS - Input Module 4-channel, 4-20 mA, 12-bit resolution
- P1-06GIO - Input Module 4-channel, 4-20 mA, 13-bit resolution
- P1-08SIM2 - Input Module 8-bit, 3.3-24 VDC

Starting at $52.00 (P1-08SIM)

Analog/Temperature I/O Modules
Analog input and output modules are available to monitor and control physical quantities, such as pressure, temperature, flow, level or any other process signal your application requires.
- P1-04AD - Input Module, 4-channel, 0-20 mA, 12-bit resolution
- P1-04AD2 - Input Module 4-channel, 0-10 VDC, 12-bit resolution
- P1-04ADL - Input Module 4-channel, 0-20 mA, 13-bit resolution
- P1-04ADL2 - Input Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04TDL - Input Module 4-channel, 4-20 mA, 12-bit resolution
- P1-04MB - Input Module 4-channel, 0-5 VDC, 16-bit resolution
- P1-04DXL - Input Module 4-channel, 0-20 VAC, 16-bit resolution
- P1-04SIM - Input Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04SIM2 - Input Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04MEM - Input Module 4-channel, 0-5 VDC, 12-bit resolution
- P1-04SIM3 - Input Module 4-channel, 0-20 mA, 12-bit resolution

Starting at $55.00 (P1-04AD)

Analog Input Modules
Analog input modules provide 4-channel, 0-20 mA, 12-bit resolution.
- P1-04AD - Input Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04AD2 - Input Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04ADL - Input Module 4-channel, 0-20 mA, 13-bit resolution
- P1-04ADL2 - Input Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04SIM - Input Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04SIM2 - Input Module 4-channel, 0-20 mA, 12-bit resolution

Starting at $55.00 (P1-04AD)

Analog Output Modules
Analog output modules provide 4-channel, 0-20 mA, 12-bit resolution.
- P1-04AD - Output Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04AD2 - Output Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04ADL - Output Module 4-channel, 0-20 mA, 13-bit resolution
- P1-04ADL2 - Output Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04SIM - Output Module 4-channel, 0-20 mA, 12-bit resolution
- P1-04SIM2 - Output Module 4-channel, 0-20 mA, 12-bit resolution

Starting at $55.00 (P1-04AD)

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-SERIAL - Serial Module, provides two 10/100 Mbps Ethernet connection
- P1AM-ETHS - Ethernet Shield, adds a single 10/100 Mbps Ethernet connection
- P1AM-SERIALS - Serial Shield, adds two 10/100 Mbps Ethernet connection
- P1AM-GPIO - MKR-pins Extension Shield, subset of MKR header pins routed to front-facing 18-pin terminal block

Starting at $95.00 (P1AM-ETH)

Click here to go now!
The P1AM 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 and serial communication 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:

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.

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.

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.

Click below or go to [http://go2adc.com/hardwarevid](http://go2adc.com/hardwarevid) to view Check out all of the hardware options in this video.
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 series, 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.

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. 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. Order yours today and get it fast with our FREE two-day shipping!

What's in the P1AM-START1?
A kit for the devoted Maker:
- (1) P1AM-100 CPU
- (1) P1AM-ETH Ethernet 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:
- (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 open-source controllers are designed to survive where others fail and we guarantee it with a two-year warranty!

Vibration/Temperature Footage:
Click here or go to http://bit.ly/shakeP1AM-100 to view

Watch as FACTS Engineering puts the controller through its paces with their in-house shaker table and temperature chamber.

For the latest prices, please check AutomationDirect.com.
Exceeding the needs of an ever-changing industry

As technology continues to evolve, 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 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 and P1AM-200 CPUs provide a C++ programming environment using the Arduino IDE. The P1AM-200 CPU also allows you to code with CircuitPython if that is preferred. Both CPUs offer supreme reliability by using industrialized hardware that can handle any job!

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.

ProductivityBlocks
Based on the Arduino 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.

ProductivityBlocks is supported by the P1AM-100 CPU and the Arduino legacy IDE (version 1.8.19 or earlier).

CircuitPython
CircuitPython is supported with the P1AM-200 CPU and is a derivative of the Python programming language built specifically for microcontrollers. Unlike the compiled code written in the Arduino IDE, CircuitPython uses a runtime interpreter which offers more flexibility with code editors and code syntax. CircuitPython can be edited using any text editor including:

- Notepad++
- Mu
- Sublime Text
- IDLE
- Thonny
- GNU Emacs

As with Arduino IDE programming, CircuitPython has a large open-source development community providing users with a wide assortment of libraries and example code for various applications.

For the latest prices, please check AutomationDirect.com.

For the latest prices, please check AutomationDirect.com.
Tons of helpful resources available anytime

AutomationDirect Community
Our technical/community forum at www.go2adc.com/P1AM is the place to go for help with your P1AM-100/200 project. There you will find:
- Links to our GitHub repositories - Library, Board Support and Mechanical
- Arduino IDE install link
- ProductivityBlocks interface download (for use w/ P1AM-100 CPU and legacy Arduino IDE only)
- Links to numerous helpful videos
- Most of all, ideas and advice from industry professionals with various backgrounds and expertise

AutomationDirect Community forum with 29,000 members and growing!
www.go2adc.com/P1AM

For the latest prices, please check AutomationDirect.com.
“But what can I do with it?”

The short answer is, “Whatever you can think of”. The P1AM CPU 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++ or CircuitPython, 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 and can easily be found online (the P1AM-200 CPU comes with several preloaded CircuitPython programs as well).

So, 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
- IoT 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!

Use it as a simple data logger
Use it to display critical information
Use it throughout your process for reliable monitoring and control

A wide variety of supported I/O

Mixing
Unloading
Filling
Capping
Packaging

For the latest prices, please check AutomationDirect.com.
Don’t take chances with maintenance costs

The cost of maintenance can quickly destroy your investment returns and your sleep!

When it comes to using a consumer-grade single-board microcontroller in an industrial environment, some say “So what if it breaks, I’ll just replace it with another inexpensive microcontroller.” While that may be an option, adding to your maintenance costs is never a winning scenario.

It’s believed that up to 20% of plant operating expense is maintenance related. That’s a good chunk and planning ahead to add to that percentage isn’t a good idea. Besides the cost of replacement parts and the labor needed, you’ll also have to consider the possibility of pulling valuable resources away from other projects, the increased probability of a failure during peak production hours, having to keep a large inventory of replacement parts on hand, and if any domino effect will occur from the failure - meaning how will the machine/system react and will other components fail as a result?

When you look at the big picture, it’s apparent that the “I’ll just swap it when it breaks” method can turn out to be quite costly. And although a consumer-grade microcontroller’s initial cost is very attractive, the savings in maintenance and downtime that you get with an industrial microcontroller blows the initial $30-or-so price difference out of the water!

### Example

You work for a package delivery service, and at your sorting facility you need to purchase an inexpensive controller to fire a single diverter arm that is feeding the new outbound conveyor. Let’s see what could happen in this industrial environment with the P1AM-100 vs. a consumer-grade microcontroller...

**Consumer-grade microcontroller**

- **Initial HW Cost**
- Diverter installed and commissioned. Diverting as expected with no issues. Initial controller and I/O hardware costs were minimal with the P1AM-100 being slightly more expensive than the consumer-grade version.

**Industrial P1AM-100**

- **Initial HW Cost**
- Controller and I/O modules are rated for operating temperatures up to 60°C/140°F so system functioned as intended without issue.

---

**Conditional Test**

- A pallet of heavy packages was sent through the sorting facility, exerting unusually high vibrations on the conveyor belts.

**Consumer-grade microcontroller**

- Vibration caused one of the shields on the controller to shake loose and the diverter was firing intermittently. Many packages were missed, causing them to be transported to wrong destination. Sort had to be extended two hours to fix the problem and trucks were very late leaving the facility. Once again additional labor costs were required, the company’s reputation was hit but the cost to reseat the shield was minimal.

**Industrial P1AM-100**

- Controller and industrial shields are rated to withstand both sinusoidal and shock vibrations so system functioned as intended without issue.

---

**Conditional Test**

- Excessive electrical noise.

**Consumer-grade microcontroller**

- Noise has caused the diverter to fire erroneously. Boxes not intended for the outbound line were diverted there and loaded on the wrong truck. The truck had to be unloaded and packages sorted again, causing deliveries to be extremely delayed. Cost of adding noise mitigation was minimal but costs to company reputation and labor expense were not.

**Industrial P1AM-100**

- Controller and I/O modules are immune to noise interference (IEC 61131-2:2017 Zone 6) so system functioned as intended without issue.

---

**Conditional Test**

- During the holiday season, the amount of packages being sorted triples and the sort runs for much longer than usual. During that time, the diverter arm fires almost continuously.

**Consumer-grade microcontroller**

- The longer runtime and excessive diverting has overheated the controller, making the output fail ON. The diverter arm stayed extended and packages were damaged as they were crushed against it. The company had to make an insurance claim to cover the losses. Company reputation took a huge hit and customers went elsewhere for their delivery needs. But the cost to replace the controller was minimal.

**Industrial P1AM-100**

- Controller and I/O modules are rated for operating temperatures up to 60°C/140°F so system functioned as intended without issue.
Consumer-grade controllers have their place, but the harsh conditions of the industrial world is not one of them. So ask yourself this simple question, when it comes to your and your company’s reputation,

**Would you rather:**

- **A low-cost open-source controller built specifically for the industrial automation field by engineers with over 25 years of service to that field, with a sound product support structure, and helpful resources that will be there for you now and in the future?**

  **Productivity**

- **A low-cost open-source controller that is intended for classrooms and hobby projects, with no long-term product support or decades of experience servicing the needs of automation professionals?**

For open-source control built for the industrial field, the choice is clear!