XLR8 ADVANTAGES

**Faster**
Hardware-accelerated functions run in a fraction of the clock cycles required to execute the same function in software.

This results in faster overall application performance.

**High-Performance**
Shorter times to complete complex tasks in hardware result in more clock cycles available for additional software functions.

This effectively improves overall computational performance.

**Scalable**
The functionality and capabilities accelerated in the FPGA hardware can be expanded and scaled for many different applications.

We are just scratching the surface of what is possible to accelerate!

**Xcelerator Blocks**
An Xcelerator Block (XB) is an optimized hardware implementation of a unique processor-intensive function. Basically, an XB is a custom piece of hardware, implemented on the same FPGA fabric and tightly integrated with the microcontroller. XBs can access the same register space and even integrate with the instructions of the microcontroller.

**Available XBs**
XLR8 ships with pre-installed XBs that target application-specific behavior, and the board can be field-updated to change the XBs implemented on the FPGA.

The default XLR8 configuration will include XBs for:
- Floating Point Math
- Servo Control
- NeoPixel Control

**XB Roadmap**
Future XBs will be implemented based on feedback from early adopters and new potential customers.

Additional XBs on our roadmap:
- Proportional-Integral-Derivative (PID) control
- Event Counters and Timers
- Quadrature Encoders/Decoders
- Pulse Width Modulation (PWM)
- Multiple UARTS
- Enhanced Analog-to-Digital Functionality
**User-Created Xcelerator Blocks**

Out of the gate, the primary goal is to provide an FPGA-based board that is a drop-in replacement for the Arduino Uno. XLR8 supports FPGA image updates via the USB port, and there is also a JTAG footprint on the board so that more advanced FPGA users could use a JTAG programmer to talk to the FPGA directly.

The microcontroller core that we have developed has been designed to be easily extendable, and Alorium Technology is actively developing the support model for users who want to create their own XB s and interface to the on-chip microcontroller.

In the near future, we plan to provide access to enough source code and documentation to make it possible for someone proficient with Verilog or VHDL and Altera’s Quartus Prime software to create their own XB s. The sky’s the limit on what can be done, and the XB s created this way can be shared with the rest of the XLR8 community.

---

**TECHNICAL SPECS**

**Physical Dimensions**
- Based on the Arduino Uno
- Matches the Uno’s physical footprint - including pin headers for attaching shields
- Mounting brackets or shields that fit Uno will also fit XLR8

**Digital I/O**
- 5V inputs
- 3.3V outputs

**Analog Inputs**
- 5V tolerant
- Op-amp circuit emulates 0-5V behavior of the ADCs on the Arduino Uno
- Correct ADC results regardless of whether it’s powered from USB or from the barrel connector
- Performance: 1 MHz
- Resolution: 12-bit sustained
- Sample Rate: 154k samples/second

---

**Specification Table**

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>ATmega328- Compatible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>3.3V with 5V I/O</td>
</tr>
<tr>
<td>Input Voltage</td>
<td>7-12V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>14</td>
</tr>
<tr>
<td>PWM Digital I/O Pins</td>
<td>6</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>6</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>32 KB</td>
</tr>
<tr>
<td>SRAM</td>
<td>2 KB</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>16/32 MHz</td>
</tr>
</tbody>
</table>

---

**Contact**
Alorium Technology
715.575.3150
www.aloriumtech.com
Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Alorium Technology:
XLR8R21M08V5U0DI   XLR8R22M08V5U0DI