CLIENT	**AdaCore**
PROJECT	**Case Study: Masten Space Systems**
OBJECTIVE	**Describe why Masten Space Systems chose to adopt the Ada/Spark language for developing the software for their XL-1 lunar lander.**

COPY EXCERPT

How Masten Space Systems is Using Ada and SPARK to Land on the Moon's South Pole

Call or write CopyEngineer to receive a PDF of the complete case study.

Or view/download it online at: https://bit.ly/AdaCore-Masten-case-study

When Masten Space Systems was awarded a NASA contract to land scientific payloads on the Moon, they knew they would need software that could be proven reliable—and proven quickly.

All the mission-critical flight control software for their XL-1 Lunar Lander had to work perfectly. And to top it all off, Masten was working with an extremely constrained schedule and budget.

That's why they chose Ada and SPARK.

The challenges of landing on the lunar south pole

In April of 2020, NASA awarded Masten a contract to transport a suite of scientific research payloads to the lunar south pole. The NASA Commercial Lunar Payload Services (CLPS) project is a $75.9 million contract that covers everything from the lab to the surface of the Moon, along with a host of payload services after landing.

Geologically, the lunar south pole is highly active and an area of great interest to scientists. The terrain is very uneven, making it a difficult place to land a spacecraft and explore with a rover. Once in lunar orbit, the XL-1 must be able to navigate to the polar region, determine where to land, and land safely, avoiding the numerous hazards in the area. It must touch down within a very narrow corridor—where the scientific interest lies, not five kilometers away. And it must do all of this autonomously.

Developing mission-critical spacecraft software to a tight schedule

Masten's lander boasts a large collection of computers, including numerous smaller computers called electronic control units (ECUs).

The ECUs are embedded systems that interface with the hardware on the spacecraft. They turn the power on and off to various components, control the engines and thrusters, and interface with the sensors.

Developing software for these small, heavily constrained ECUs presents a number of challenges, according to Abhimanyu Ghosh, the avionic software engineer who is leading Masten's ECU software development effort for the XL-1.

First, memory and power are scarce resources on the XL-1, as they are on most spacecraft. To minimize resource consumption, the ECU software will run in a "bare metal" configuration, that is, without an operating system or other facilities that support the application code.

Second, the software must be easy to maintain. Therefore, it must be easy to understand for both technical and non-technical team members…