1.	Record Nr.	UNINA9910411926503321
	Autore	Coburn Joseph
	Titolo	Build Your Own Car Dashboard with a Raspberry Pi : Practical Projects to Build Your Own Smart Car / / by Joseph Coburn
	Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2020
	ISBN	1-4842-6080-5
	Edizione	[1st ed. 2020.]
	Descrizione fisica	1 online resource (XIX, 306 p. 101 illus.)
	Collana	Technology in Action Series
	Disciplina	005.3
	Soggetti	Makerspaces Maker
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Nota di bibliografia	Includes bibliographical references.
	Nota di contenuto	Chapter 1 - Raspberry Pi History -- Chapter 2 - Software Development Primer -- Chapter 3 - Project Overview -- Chapter 4 - Development Environment_Configuration -- Chapter 5 - Raspberry Pi Configuration -- Chapter 6 - Getting Started with Flask -- Chapter 7: Temperature Monitoring -- Chapter 8 - Boot Sensor -- Chapter 9 - Light Sensor -- Chapter 10: Fog and Reverse Sensors -- Chapter 11: Reversing Camera -- Chapter 12: Reversing Beeper -- Chapter 13: Distance Sensor -- Chapter 14: System Polishing. .
	Sommario/riassunto	Create your own car engine control unit (ECU) with a simple Raspberry PI while building the necessary skills to produce future more advanced projects. Once you've worked through the projects in this book, you'll have a smart car and the coding knowledge needed to develop advanced hardware and software projects. Start by understanding how the Pi works, and move on to how to build hardware projects, use the GPIO pins, and install the system. Then add to that a solid understanding of software development principles and best practices, along with a good grasp of Python (v3.6+) and Python/software best practices. More than just how to code in Python, you'll learn what it takes to write production grade software, defensive code, testing, deployments, version control, and more. Internalize industry best practices while going further with valuable software development techniques such as defensive programming. The concepts introduced are essential to ensuring that software can function under unexpected

circumstances. Can you imagine what would happen if your mobile phone could not cope with a call from an unknown number, or you had to set you microwave in increments of 6 seconds? While testing avoids edge cases such as these, defensive programming is one of the building blocks of software development. You will: Hone test driven development in Python skills Debug software and hardware project installations Work with the GPIO ports of the Pi to feed your software real-world hardware information.