1.

Record Nr.	UNINA9910555117703321
Autore	Unsalan Cem
Titolo	Embedded digital control with microcontrollers : implementation with C and Python / / Unsalan, Duygun E. Barkana, and H. Deniz Gurhan
Pubbl/distr/stampa	Hoboken, New Jersey : , : John Wiley & Sons, Incorporated, , [2021] ©2021
ISBN	1-119-57655-5 1-119-57660-1 1-119-57658-X
Descrizione fisica	1 online resource (371 pages) : illustrations
Collana	Wiley - IEEE Ser.
Disciplina	629.89
Soggetti	C (Computer program language) Digital control systems Microcontrollers
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Cover -- Title Page -- Copyright -- Contents -- Preface -- About the Companion Website -- Chapter 1 Introduction -- 1.1 What is a System? -- 1.2 What is a Control System? -- 1.3 About the Book -- Chapter 2 Hardware to be Used in the Book -- 2.1 The STM32 Board -- 2.1.1 General Information -- 2.1.2 Pin Layout -- 2.1.3 Powering and Programming the Board -- 2.2 The STM32 Microcontroller -- 2.2.1 Central Processing Unit -- 2.2.2 Memory -- 2.2.3 Input and Output Ports -- 2.2.4 Timer Modules -- 2.2.5 ADC and DAC Modules -- 2.2.6 Digital Communication Modules -- 2.3 System and Sensors to be Used Throughout the Book -- 2.3.1 The DC Motor -- 2.3.1.1 Properties of the DC Motor -- 2.3.1.2 Pin Layout -- 2.3.1.3 Power Settings -- 2.3.2 The DC Motor Drive Expansion Board -- 2.3.3 Encoder -- 2.3.4 The FT232 Module -- 2.4 Systems and Sensors to be Used in Advanced Applications -- 2.4.1 Systems -- 2.4.2 Sensors -- 2.5 Summary -- Problems -- Chapter 3 Software to be Used in the Book -- 3.1 Python on PC -- 3.1.1 Basic Operations -- 3.1.2 Array and Matrix Operations -- 3.1.3 Loop Operations -- 3.1.4 Conditional Statements -- 3.1.5 Function Definition and Usage -- 3.1.6 File Operations -- 3.1.7 Python

Control Systems Library -- 3.2 MicroPython on the STM32 Microcontroller -- 3.2.1 Setting up MicroPython -- 3.2.2 Running MicroPython -- 3.2.3 Reaching Microcontroller Hardware -- 3.2.3.1 Input and Output Ports -- 3.2.3.2 Timers -- 3.2.3.3 ADC -- 3.2.3.4 DAC -- 3.2.3.5 UART -- 3.2.4 MicroPython Control Systems Library -- 3.3 C on the STM32 Microcontroller -- 3.3.1 Creating a New Project in Mbed Studio -- 3.3.2 Building and Executing the Code -- 3.3.3 Reaching Microcontroller Hardware -- 3.3.3.1 Input and Output Ports -- 3.3.3.2 Timers -- 3.3.3.3 ADC -- 3.3.3.4 DAC -- 3.3.3.5 UART -- 3.3.4 C Control Systems Library -- 3.4 Application: Running the DC Motor.

3.4.1 Hardware Setup -- 3.4.2 Procedure -- 3.4.3 C Code for the System -- 3.4.4 Python Code for the System -- 3.4.5 Observing Outputs -- 3.5 Summary -- Problems -- Chapter 4 Fundamentals of Digital Control -- 4.1 Digital Signals -- 4.1.1 Mathematical Definition -- 4.1.2 Representing Digital Signals in Code -- 4.1.2.1 Representation in Python -- 4.1.2.2 Representation in C -- 4.1.3 Standard Digital Signals -- 4.1.3.1 Unit Pulse Signal -- 4.1.3.2 Step Signal -- 4.1.3.3 Ramp Signal -- 4.1.3.4 Parabolic Signal -- 4.1.3.5 Exponential Signal -- 4.1.3.6 Sinusoidal Signal -- 4.1.3.7 Damped Sinusoidal Signal -- 4.1.3.8 Rectangular Signal -- 4.1.3.9 Sum of Sinusoids Signal -- 4.1.3.10 Sweep Signal -- 4.1.3.11 Random Signal -- 4.2 Digital Systems -- 4.2.1 Mathematical Definition -- 4.2.2 Representing Digital Systems in Code -- 4.2.2.1 Representation in Python -- 4.2.2.2 Representation in C -- 4.2.3 Digital System Properties -- 4.2.3.1 Stability -- 4.2.3.2 Linearity -- 4.2.3.3 TimeInvariance -- 4.3 Linear and TimeInvariant Systems -- 4.3.1 Mathematical Definition -- 4.3.2 LTI Systems and ConstantCoefficient Difference Equations -- 4.3.3 Representing LTI Systems in Code -- 4.3.3.1 MicroPython Control Systems Library Usage -- 4.3.3.2 C Control Systems Library Usage -- 4.3.3.3 Python Control Systems Library Usage -- 4.3.4 Connecting LTI Systems -- 4.3.4.1 Series Connection -- 4.3.4.2 Parallel Connection -- 4.3.4.3 Feedback Connection -- 4.4 The zTransform and Its Inverse -- 4.4.1 Definition of the zTransform -- 4.4.2 Calculating the z Transform in Python -- 4.4.3 Definition of the Inverse zTransform -- 4.4.4 Calculating the Inverse zTransform in Python -- 4.5 The z Transform and LTI Systems -- 4.5.1 Associating Difference Equation and Impulse Response of an LTI System -- 4.5.2 Stability Analysis of an LTI System using zTransform.

4.5.3 Stability Analysis of an LTI System in Code -- 4.6 Application I: Acquiring Digital Signals from the Microcontroller, Processing Offline Data -- 4.6.1 Hardware Setup -- 4.6.2 Procedure -- 4.6.3 C Code for the System -- 4.6.4 Python Code for the System -- 4.6.5 Observing Outputs -- 4.7 Application II: Acquiring Digital Signals from the Microcontroller, Processing RealTime Data -- 4.7.1 Hardware Setup -- 4.7.2 Procedure -- 4.7.3 C Code for the System -- 4.7.4 Python Code for the System -- 4.7.5 Observing Outputs -- 4.8 Summary -- Problems -- Chapter 5 Conversion Between Analog and Digital Forms -- 5.1 Converting an Analog Signal to Digital Form -- 5.1.1 Mathematical Derivation of ADC -- 5.1.2 ADC in Code -- 5.2 Converting a Digital Signal to Analog Form -- 5.2.1 Mathematical Derivation of DAC -- 5.2.2 DAC in Code -- 5.3 Representing an Analog System in Digital Form -- 5.3.1 PoleZero Matching Method -- 5.3.2 ZeroOrder Hold Equivalent -- 5.3.3 Bilinear Transformation -- 5.4 Application: Exciting and Simulating the RC Filter -- 5.4.1 Hardware Setup -- 5.4.2 Procedure -- 5.4.3 C Code for the System -- 5.4.4 Python Code for the System -- 5.4.5 Observing Outputs -- 5.5 Summary -- Problems -- Chapter 6 Constructing Transfer Function of a

System -- 6.1 Transfer Function from Mathematical Modeling -- 6.1.1 Fundamental Electrical and Mechanical Components -- 6.1.2 Constructing the Differential Equation Representing the System -- 6.1.3 From Differential Equation to Transfer Function -- 6.2 Transfer Function from System Identification in Time Domain -- 6.2.1 Theoretical Background -- 6.2.2 The Procedure -- 6.2.3 Data Acquisition by the STM32 Microcontroller -- 6.2.4 System Identification in Time Domain by MATLAB -- 6.3 Transfer Function from System Identification in Frequency Domain -- 6.3.1 Theoretical Background -- 6.3.2 The Procedure.

6.3.3 System Identification in Frequency Domain by MATLAB -- 6.4 Application: Obtaining Transfer Function of the DC Motor -- 6.4.1 Mathematical Modeling -- 6.4.2 System Identification in Time Domain -- 6.4.3 System Identification in Frequency Domain -- 6.5 Summary -- Problems -- Chapter 7 Transfer Function Based Control System Analysis -- 7.1 Analyzing System Performance -- 7.1.1 Time Domain Analysis -- 7.1.1.1 Transient Response -- 7.1.1.2 SteadyState Error -- 7.1.2 Frequency Domain Analysis -- 7.1.3 Complex Plane Analysis -- 7.1.3.1 RootLocus Plot -- 7.1.3.2 Nyquist Plot -- 7.2 The Effect of OpenLoop Control on System Performance -- 7.2.1 What is OpenLoop Control? -- 7.2.2 Improving the System Performance by OpenLoop Control -- 7.3 The Effect of ClosedLoop Control on System Performance -- 7.3.1 What is ClosedLoop Control? -- 7.3.2 Improving the System Performance by ClosedLoop Control -- 7.4 Application: Adding OpenLoop Digital Controller to the DC Motor -- 7.4.1 Hardware Setup -- 7.4.2 Procedure -- 7.4.3 C Code for the System -- 7.4.4 Python Code for the System -- 7.4.5 Observing Outputs -- 7.5 Summary -- Problems -- Chapter 8 Transfer Function Based Controller Design -- 8.1 PID Controller Structure -- 8.1.1 The P Controller -- 8.1.2 The PI Controller -- 8.1.3 The PID Controller -- 8.1.4 Parameter Tuning Methods -- 8.1.4.1 The Ziegler-Nichols Method -- 8.1.4.2 The Cohen-Coon Method -- 8.1.4.3 The Chien-Hrones-Reswick Method -- 8.2 PID Controller Design in Python -- 8.2.1 Parameter Tuning -- 8.2.2 Controller Design -- 8.2.2.1 P Controller -- 8.2.2.2 PI Controller -- 8.2.2.3 PID Controller -- 8.2.3 Comparison of the Designed P, PI, and PID Controllers -- 8.3 Lag-Lead Controller Structure -- 8.3.1 Lag Controller -- 8.3.2 Lead Controller -- 8.3.3 Lag-Lead Controller -- 8.4 Lag-Lead Controller Design in MATLAB.

8.4.1 Control System Designer Tool -- 8.4.2 Controller Design in Complex Plane -- 8.4.2.1 Lag Controller -- 8.4.2.2 Lead Controller -- 8.4.2.3 Lag-Lead Controller -- 8.4.2.4 Comparison of the Designed Lag, Lead, and Lag-Lead Controllers -- 8.4.3 Controller Design in Frequency Domain -- 8.4.3.1 Lag Controller -- 8.4.3.2 Lead Controller -- 8.4.3.3 Lag-Lead Controller -- 8.4.3.4 Comparison of the Designed Lag, Lead, and Lag-Lead Controllers -- 8.5 Application: Adding Closed Loop Digital Controller to the DC Motor -- 8.5.1 Hardware Setup -- 8.5.2 Procedure -- 8.5.3 C Code for the System -- 8.5.4 Python Code for the System -- 8.5.5 Observing Outputs -- 8.6 Summary -- Problems -- Chapter 9 Statespace Based Control System Analysis -- 9.1 Statespace Approach -- 9.1.1 Definition of the State -- 9.1.2 Why Statespace Representation? -- 9.2 Statespace Equations Representing an LTI System -- 9.2.1 Continuoustime Statespace Equations -- 9.2.2 Discretetime Statespace Equations -- 9.2.3 Representing Discrete time Statespace Equations in Code Form -- 9.3 Conversion Between Statespace and Transfer Function Representations -- 9.3.1 From Transfer Function to Statespace Equations -- 9.3.2 From Statespace Equations to Transfer Function -- 9.4 Properties of the System from its Statespace Representation -- 9.4.1 Time Domain Analysis -- 9.4.2

Stability -- 9.4.3 Controllability -- 9.4.4 Observability -- 9.5 Application: Observing States of the DC Motor in Time -- 9.5.1 Hardware Setup -- 9.5.2 Procedure -- 9.5.3 C Code for the System -- 9.5.4 Python Code for the System -- 9.5.5 Observing Outputs -- 9.6 Summary -- Problems -- Chapter 10 Statespace Based Controller Design -- 10.1 General Layout -- 10.1.1 Control Based on State Values -- 10.1.2 Regulator Structure -- 10.1.3 Controller Structure -- 10.1.4 What if States Cannot be Measured Directly?.

10.2 Regulator and Controller Design via Pole Placement.