This document covers analog-to-digital conversion (ADC) in embedded systems, beginning with a review of C programming and logic operations. It introduces AVR microcontroller architecture and programming, focusing on its ADC, interrupt, timing, and serial communication subsystems. The core concept differentiates analog signals (continuous values) from digital signals (discrete values), highlighting the necessity of ADC for digital systems interacting with the analog world (e.g., voice communication). ADC involves sampling (taking signal snapshots), quantization (approximating sample values to discrete levels), and encoding. Optimal sampling, determined by the Nyquist rate (2fm, where fm is the maximum input frequency), prevents aliasing (signal distortion). Quantization introduces error, reduced by increasing the number of quantization levels (determined by the number of bits). Higher resolution (smaller step size between levels) requires more bits, increasing the data rate (bits per second). The example of a telephone system illustrates how increased bit depth for higher resolution .significantly impacts data rate and system capacity