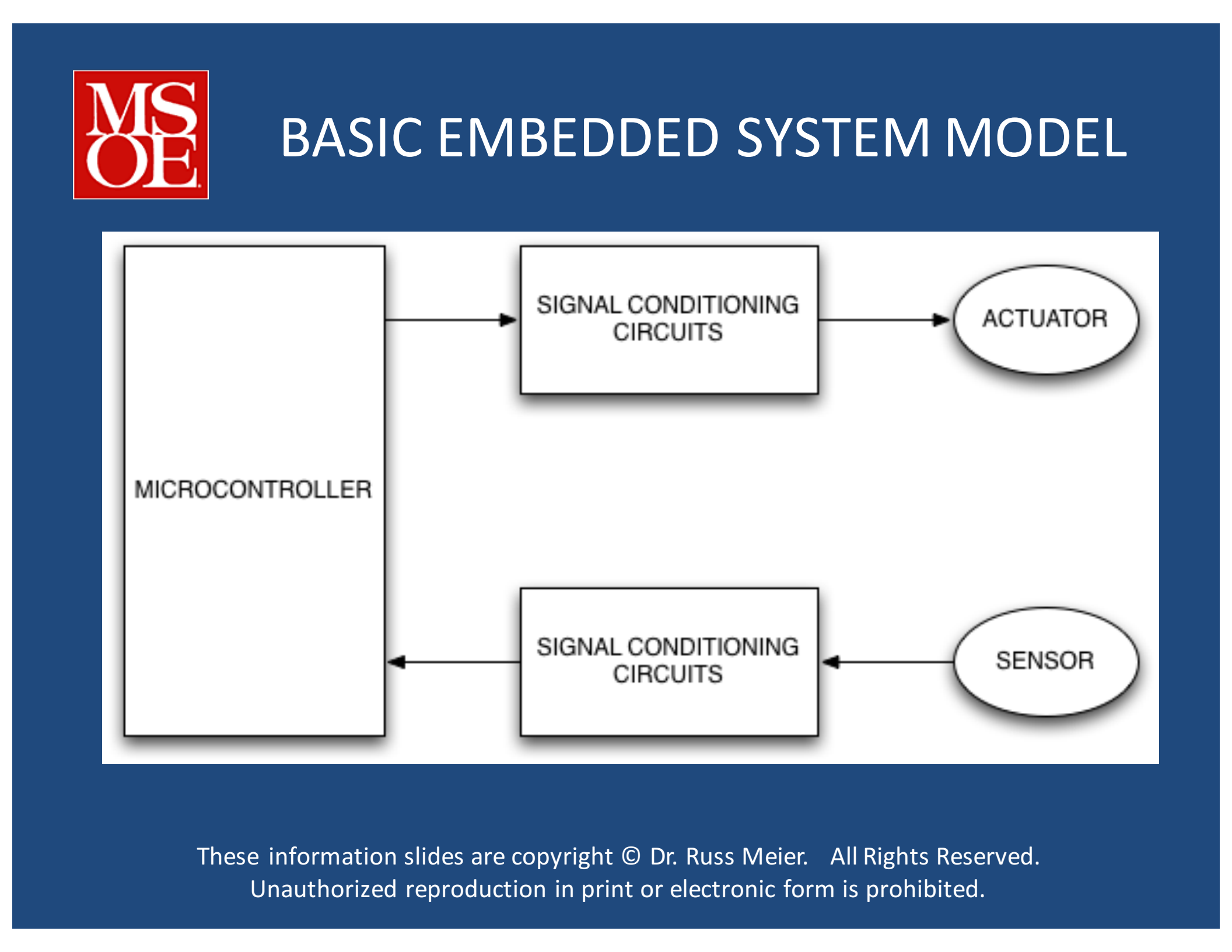
C LANGUAGE FUNDAMENTALS FOR EMBEDDED SYSTEMS

Dr. Russ Meier

MSOE



BASIC EMBEDDED SYSTEM MODEL

SENSORS

- Transducers convert energy between forms.
- Sensors are transducers completing P→E
- Example sensors
  - Microphone
  - Photoreceiver
  - Buttons
  - Slide switches
  - Flowmeter

ACTUATORS

- Actuators are transducers completing E→P
- Examples
  - Speaker or buzzer
  - Motor
  - LED
  - LCD Panel

SIGNAL CONDITIONING CIRCUIT

- Most sensed environmental signals are small
- Small signals don't use ADC full-scale range
- Signal conditioning circuits range V or I
- Examples
  - Amplifiers
  - Filters
  - DAC

PROGRAMMING PARADIGMS

- Procedural programming
- Object oriented programming
- Functional programming
- Logic programming

- They differ in approach to data and calculation

PROCEDURAL PARADIGM

- Focuses on the calculation and control.
- The job of the programmer is to determine the set of tasks required to get the work done.
- The focus is not on the data.
- The focus is on the procedures and functions.
- Data is passed into procedures and functions.
- Data access is not strongly protected.

OBJECT PARADIGM

- Focuses on the data.
- The job of the programmer is to determine the data objects required to get the job done.
- The focus is on the data object.
- Data objects define their manipulation functions.
- Data objects are strongly protected.

C HISTORY

- Timeline
  - Developed at AT&T Bell Labs
  - Part of the UNIX project
  - Five year period of development (1969-1973)
  - Dennis Ritchie is credited as the author of C
  - Brian Kernigan and Ritchie publish the first authoritative C reference in 1978 (K&R C)

C HISTORY

- General-purpose programming language
  - procedural language without user defined objects
  - original use was OS software and device drivers
  - C has found widespread use for applications as well
  - C is currently the most popular programming language among code developers (www.tiobe.com)

WHY IS C POPULAR?

- Small set of keywords
- Strongly typed
- Ability to access any memory byte
- Portability
- Standard library of functions
- Efficient assembly language because of simplicity

STANDARDIZATION

- C is available for nearly every architecture
- Standardization became very important:
  - cross-platform compilation
  - software engineering paradigms
  - guide compiler writers

STANDARIZATION

- American National Standards Institute
  - Released the first standard in 1989
  - ANSI 1989 commonly called C89
- International Standards Organization
  - Adopted the ANSI version in 1990
  - ISO 1990 commonly called C90

STANDARDIZATION

- Revisions were made to the standard in 1999 (ANSI/ISO C99)
  - new data types
  - inline functions
  - variable-length arrays
  - line comments (//)
- Note that GCC is not yet fully C99 compliant.

PROGRAMMING CONCEPTS

- Control flow
  - Control unit moves data to arithmetic circuits
  - Calculations are completed
  - Results are stored in memory
  - Keywords: operators, comparison, control flow
  - C Keyword examples: if, case, while

PROGRAMMING CONCEPTS

- Data typing
  - Declares names of variables
  - Defines type of storage and number of bits
  - Keywords: type specifiers and type qualifiers
  - C Keyword Examples: int, char, const

PROGRAMMING CONCEPTS

- Memory control
  - Defines where variables are stored
  - Defines access restrictions and visibility
  - Keywords: storage class
  - C Keyword examples: register, extern, static

C KEYWORDS

- C has a small set of reserved keywords
- All C keywords can be classified as:
  - flow control
  - type specifier
  - type qualifier
  - storage class
  - operator

C FLOW CONTROL KEYWORDS

- break
- case
- continue
- default
- do
- else
- for
- goto
- if
- return
- switch
- while

C TYPE SPECIFIER KEYWORDS

- char
- double
- enum
- float
- int
- long
- short
- signed
- struct
- typedef
- union
- unsigned
- void

C TYPE QUALIFIER KEYWORDS

- const
  - constant values
  - should not be changed
  - enforced on variables
  - not enforced on pointers
- volatile
  - used in embedded systems
  - used in systems software
  - asynchronous modification
  - do not optimize away

C STORAGE CLASS KEYWORDS

- auto
- extern
- static
- register

WHAT IS STORAGE CLASS

- where is the data
- when is it allocated
- when is it deallocated
- what can see it

C STORAGE CLASSES

Keyword	Where	Allocated	De-allocated	Initialized
auto	stack	block entry	block exit	uninitialized
extern				
register	stack or register	block entry	block exit	uninitialized
static	data memory	compilation	program exit	zero