International Workshop on *Design Principles for Next Generation Embedded Computing Systems*

Amit Kumar Singh

School of Computer Science and Electronic Engineering

University of Essex

United Kingdom

W: http://aksingh.co.uk/ E: a.k.singh@essex.ac.uk

Embedded and Intelligent Systems Laboratory @Essex

- **Research Areas**
  - **Computer vision and embedded AI for real world problems**
  - **Advanced embedded systems for several applications, e.g. automotive, robotics, image processing and IoT**
  - **Technologies to improve security, reliability, power and performance**
- **Many academics and researchers**
- **Several funded research projects**

https://eis.essex.ac.uk/

Outline

- **Introduction to embedded systems**
- **Multi-core based next generation embedded systems**
- **Applications execution on next generation embedded systems**
- **Important metrics for embedded systems**
- **Design-time optimisation principles for various metrics**
- **Run-time time optimisation principles for various metric**
- **Online programming demonstration for a multi-core based embedded system**
- **Envisioned future and design challenges for next generation embedded systems**

Where are you located today? (You can answer as: Guwahati-India, Delhi-India, London-UK, etc.

What is your name and designation? (You can answer as: Amit-Associate Professor, Alex-Embedded Engineer, Adam-BTech Student, etc.)

What are your interests? (You can answer as: Embedded Systems, C Programming, etc.)

Introduction to Embedded Systems

Respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join, then **A, B, C, or D**

What is an Embedded System?

Any computer system that is not perceived as general purpose computer systems such as PCs or supercomputers.	**A**
Any computer system that is hidden (embedded) inside another system.	**B**
Any computer system that performs only a set of dedicated tasks.	**C**
Any of the above can be a definition of embedded system.	**D**

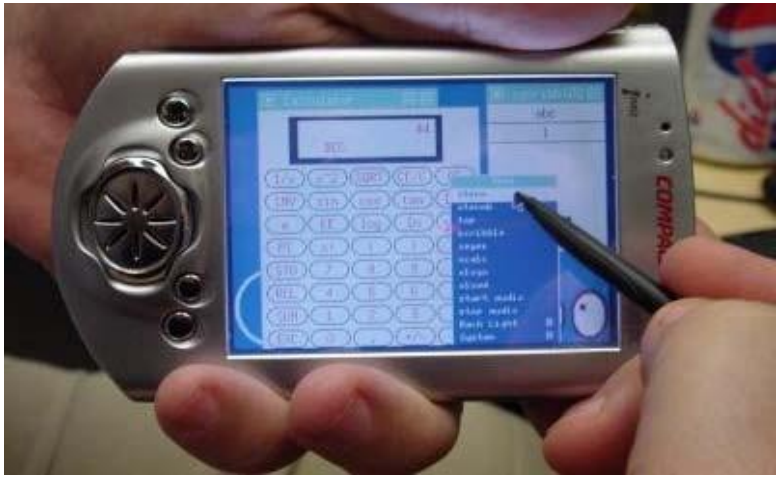
What is an Embedded System?

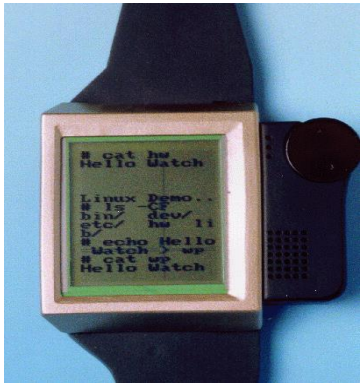
"Embedded systems are dedicated computing systems that are hidden inside other systems and are typically not perceived as general purpose computing systems."

They can be processor based or dedicated circuits

Examples of processor-based embedded systems range from watches to nuclear missiles and intelligent weapons.

Examples of Embedded Systems





Usages of Embedded Systems

Less than 2% of microprocessors manufactured each year are used in general purpose computer systems!

The majority are used in some form of embedded system!

Usages of Embedded Systems

- A car may have few microprocessors for engine management, anti-lock brakes, etc.

- More with autonomous cars

Usages of Embedded Systems

- A washing machine may have a microprocessor based control unit for wash program control, timing, spin speed, etc.

- More microprocessor based control units with advanced washing machines.

Usages of Embedded Systems

- Many toys and domestic appliances use microprocessor control.

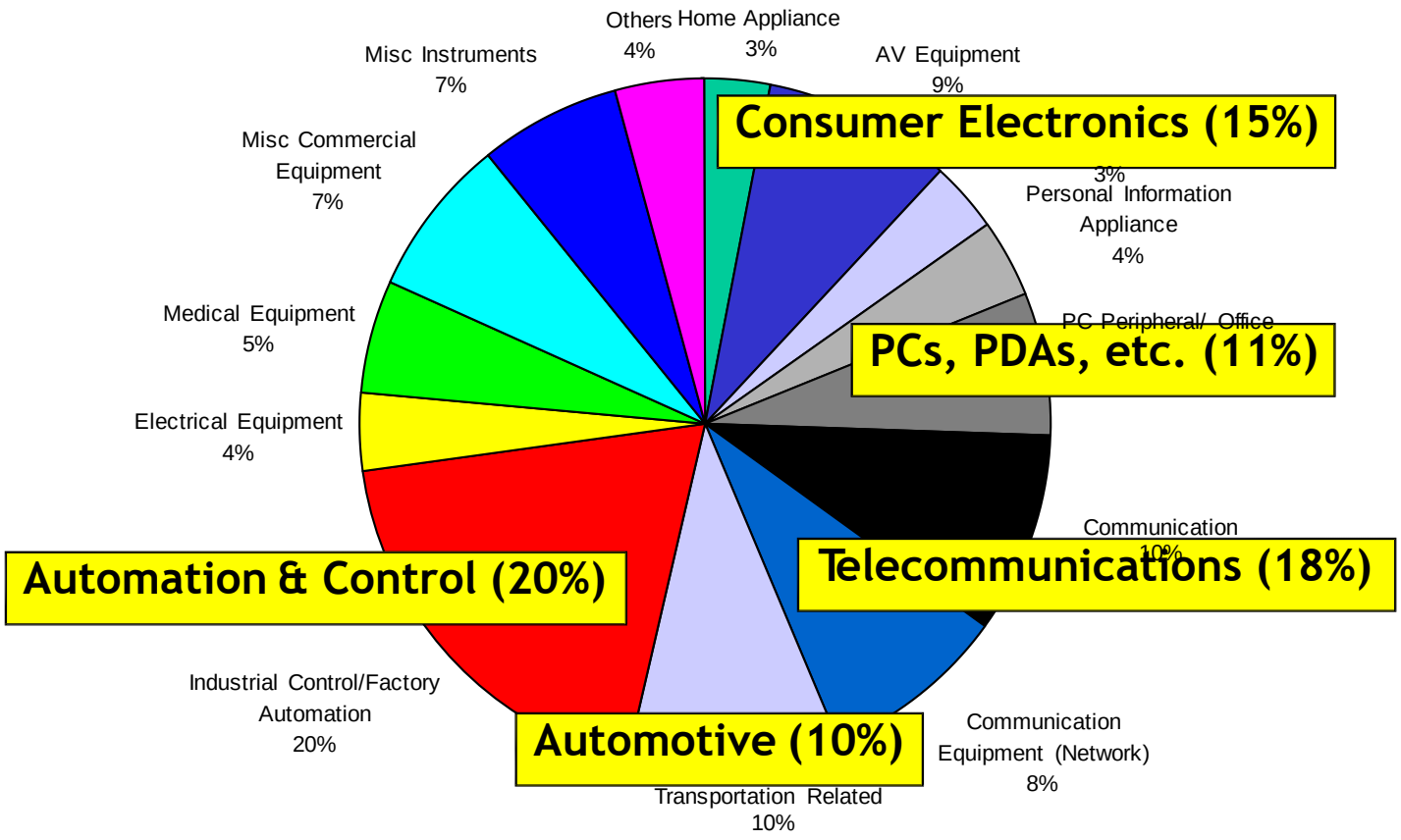
Usages of Embedded Systems

- A TV remote controller or the PC keyboard has a microprocessor inside.

Usages of Embedded Systems

- Mobile phones contain more computing power than old desktop PC.

Classes of Embedded Systems

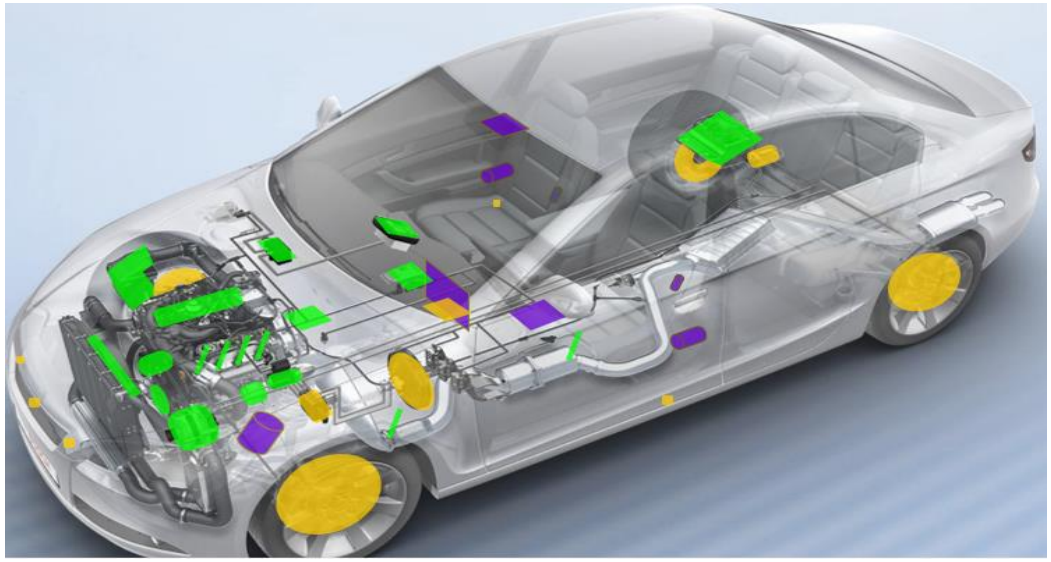


Features of Embedded Systems

- Flexible functionality.
- Real-time operation.
- Low manufacturing cost.
- Often designed to tight deadlines.
  - Often designed by a small team of designers.
  - Often must meet tight deadlines (6 months market window is common.)

Trends in Embedded Systems

- The rising numbers
  - Modern automotive can easily have 80 to 100 processors on board



Trends in Embedded Systems

- The rising complexity
  - Connectivity using Internet, Wireless communications
  - 32 bits processors in place of 4 or 8 bits processors
  - Interoperability, Security, Scalability, etc. required

Trends in Embedded Systems

- The rising impact and importance
  - Usage in safety critical or mission critical applications

Trends in Embedded Systems

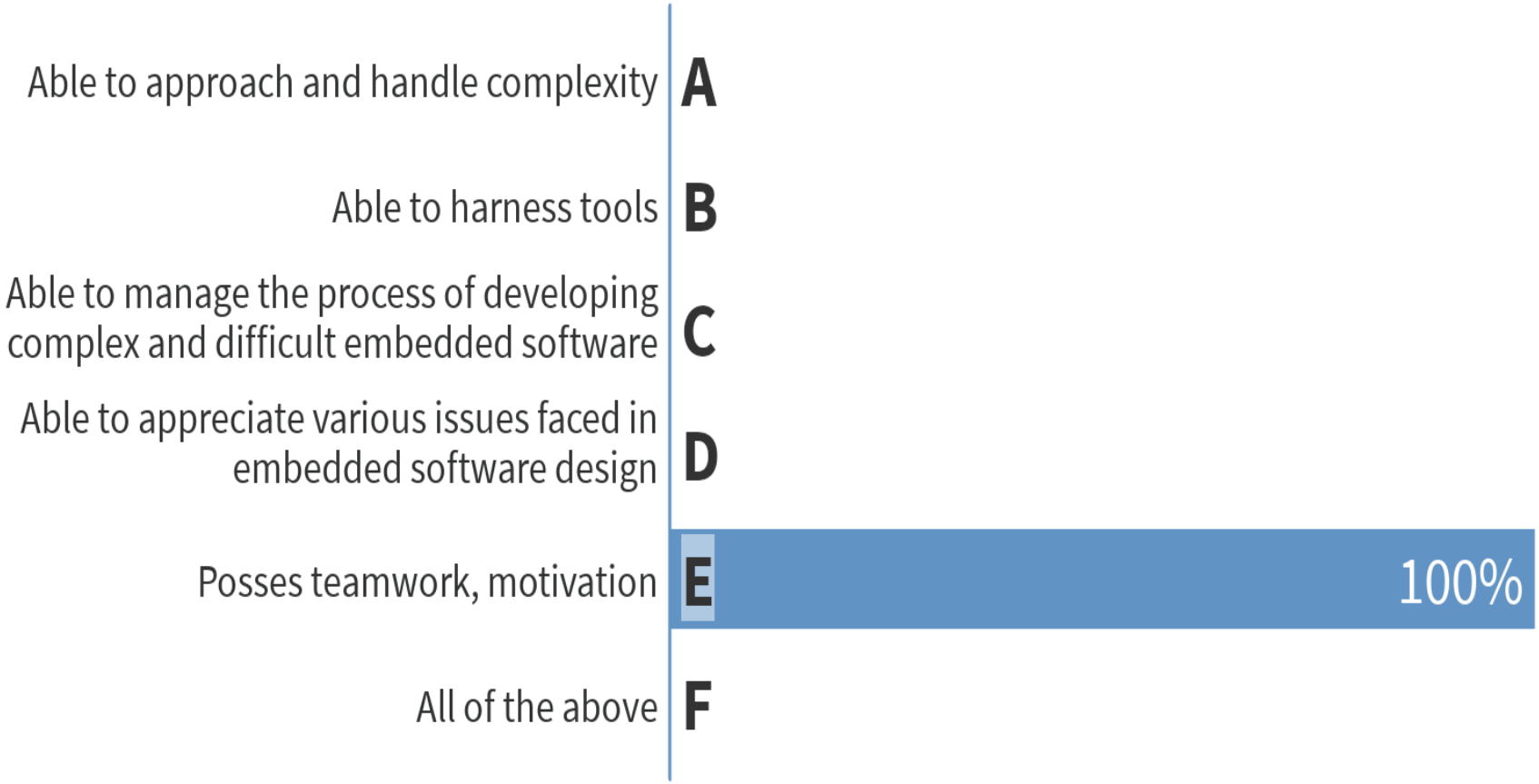
- Rising need to get the embedded software right, the first time and all the time
  - Failures can be fatal
  - re-engineering and redesign can be very expensive

Embedded Systems Software Engineer

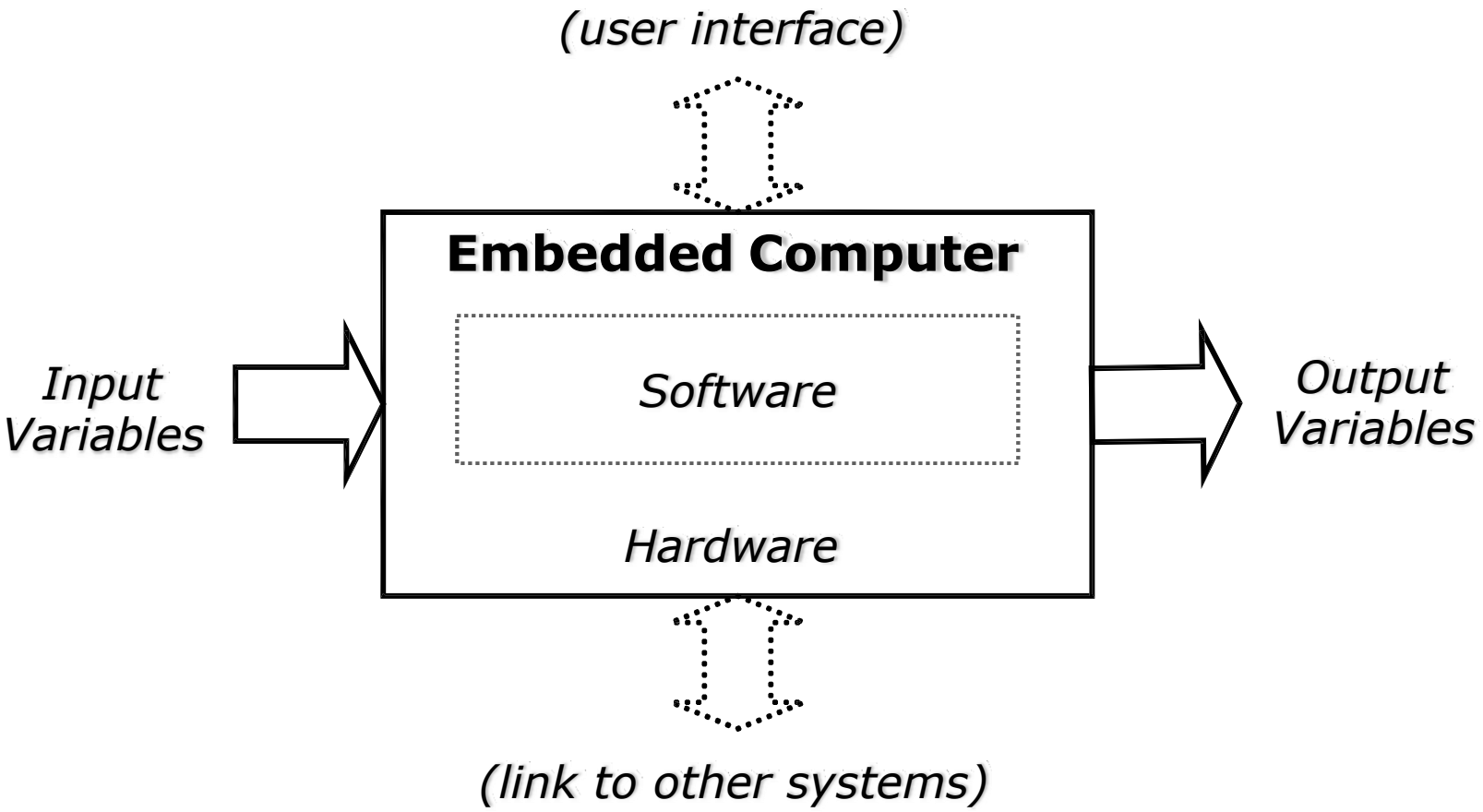
- Need to have several desired traits to handle the trends and features of Embedded Systems

- Let’s see it with an easy question!

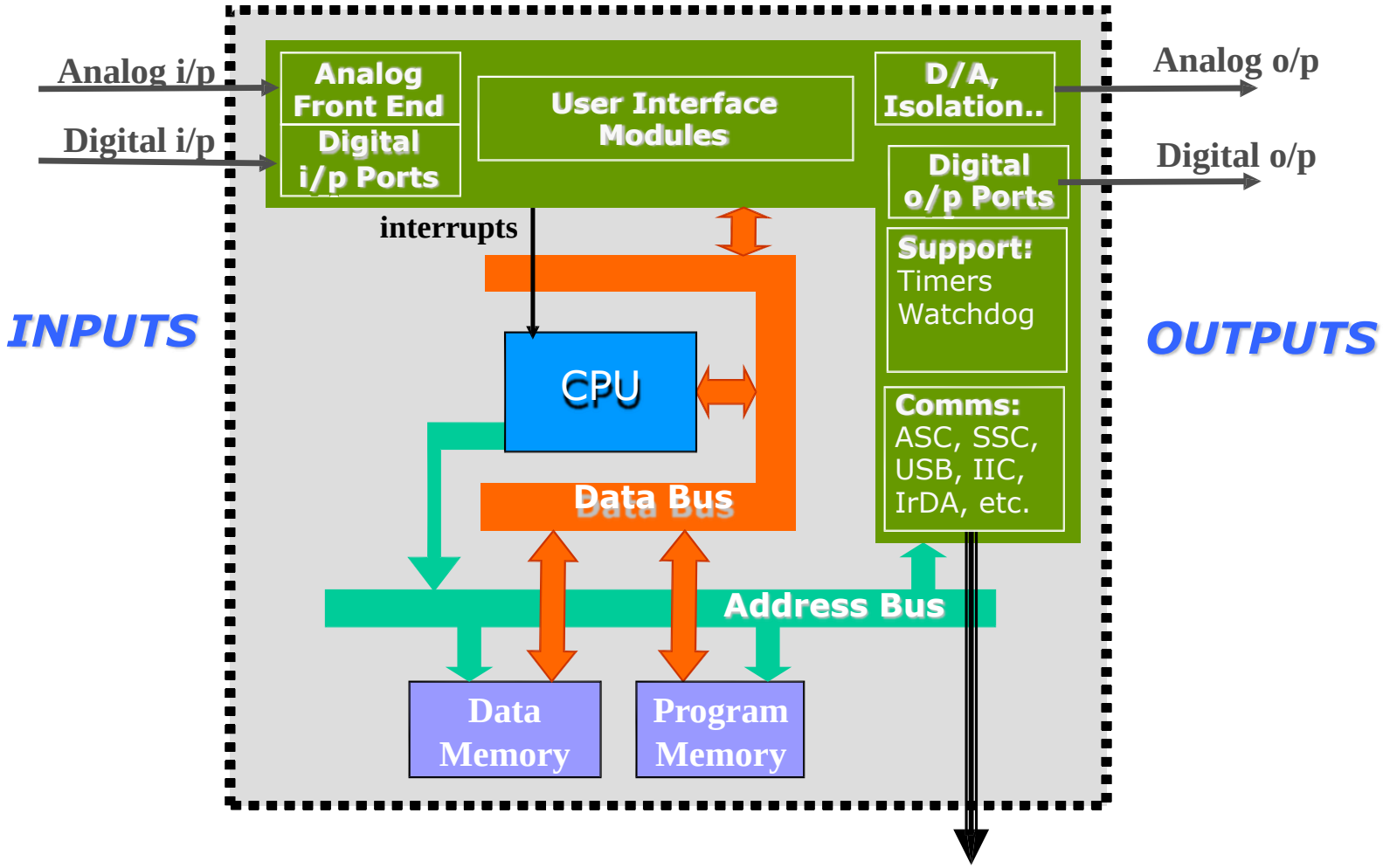
What are the desired traits of an embedded systems software engineer?



An Embedded Computer



User Interface



Respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join, then **A, B, C, or D**

Consider your smartphone as an embedded computer, what will be Input?

Incoming call	**A**
Receive button or swipe	**B**
Listening to the call	**C**
Diverting to other system	**D**

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

When poll is active, respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join

Consider your smartphone as an embedded computer, what will be Output?

Incoming call

Receive button or swipe

Listening to the call

Diverting to other system

When poll is active, respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join

Consider your smartphone as an embedded computer, what will be User Interface?

Incoming call

Receive button or swipe

Listening to the call

Diverting to other system

Respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join, then **A, B, C, or D**

Consider your smartphone as an embedded computer, what will be Links to other system?

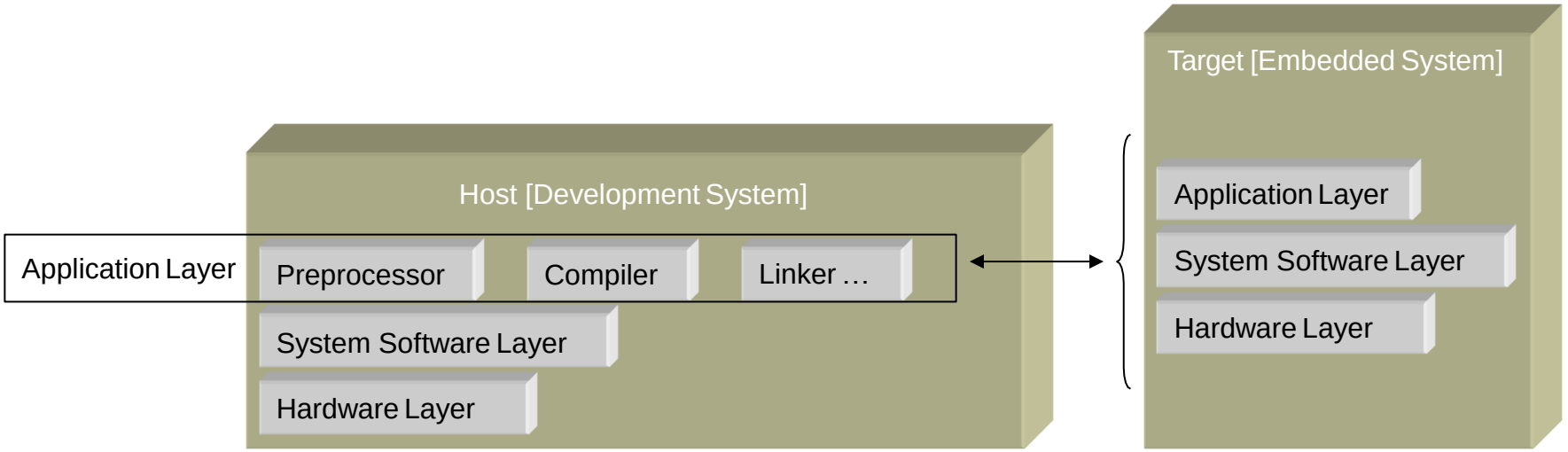
Incoming call	**A**
Receive button or swipe	**B**
Listening to the call	**C**
Diverting to other system	**D**

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Typical Embedded Software Development

Typical Embedded Software Development

- Cross compilation of the software for the Target system is usually performed on a Host system
- Code can then be downloaded onto the Target system



When poll is active, respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join

Why such a compilation procedure, i.e. cross compilation of the software for the target system on a Host system and then downloading onto the Target system?

Due to limited resources on the target system

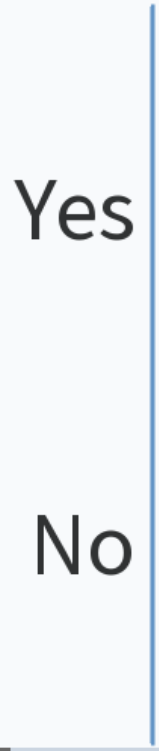
Due to conventional software development procedure

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

When poll is active, respond at **pollev.com/amitsingh510**

Text **AMITSINGH510** to **22333** once to join

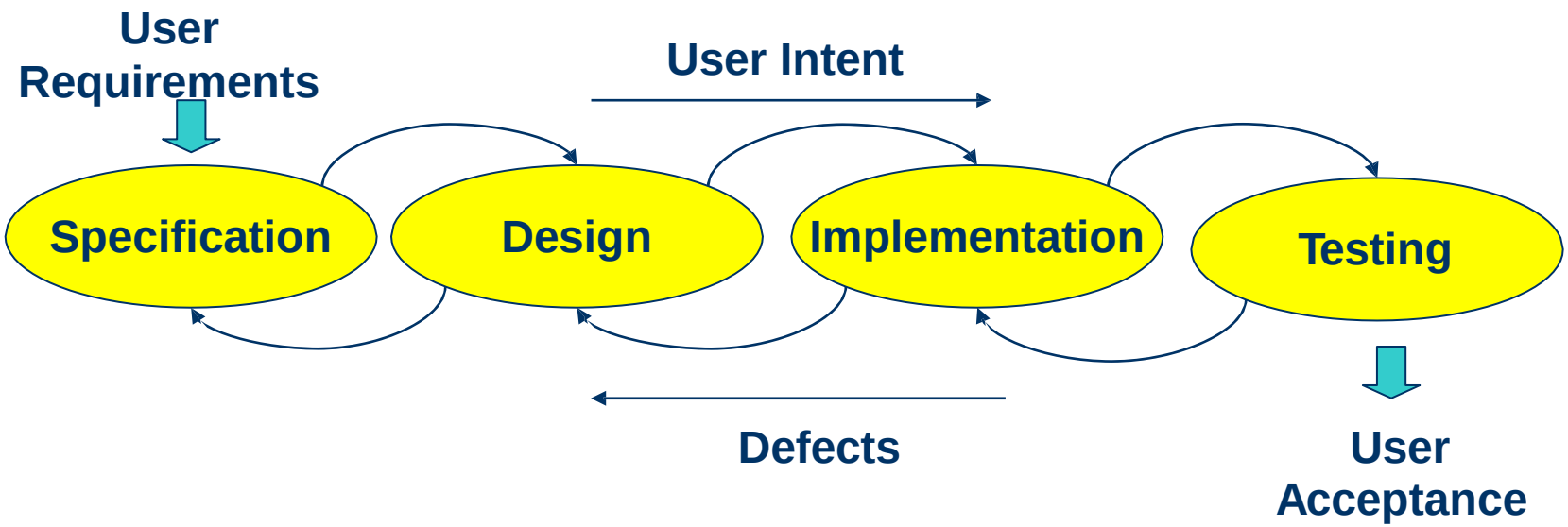
Would you follow the cross-compilation for compiling a program for your desktop?



Lifecycle of an Embedded System

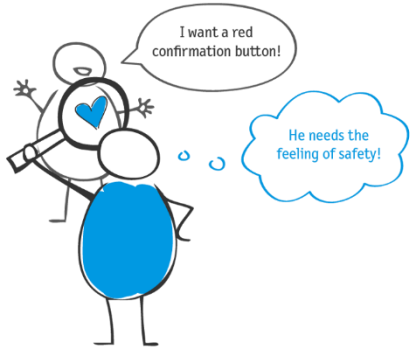
Lifecycle of an Embedded System

- Similar to typical product development lifecycle
  - Requirements
  - Design
  - Implementation
  - Integration and Testing
  - Maintenance and Upgrading



Lifecycle: Requirements

- Objective
  - ➢ To capture the user intent and translate into expectations for the system
- Challenges
  - ➢ Users don’t know what they want
    - No visualization, No domain knowledge
  - ➢ Developers don’t know what they are designing for
    - No norms, No precedence, Highly customized or specialized applications, No prototype or evolving prototype



Lifecycle: Design

- Objective
  - To create a solution (design) that best fits the application within the constraints given
- Challenges
  - Resource Constraints
    - Processing power (CPU frequency, turning of unnecessary logic, reducing memory accesses), memory size, battery life, etc. -> cost limitations
  - Real Time Constraints
    - Programming languages and operating systems selected must be capable of handling real time requirements (order of usec)
    - Faster hardware or cleverer software
    - Concurrency is required
  - Safety Constraints
    - Need to design safety into the system, not an afterthought
    - Fail-safe, fail operational
  - Marketability and scalability

Lifecycle: Implementation

- Objective
  - To implement the solution selected, optimizing for performance and achieving robustness
- Challenges
  - Optimization
    - Squeezing out the last drop of computing power
  - Robust software
    - Embedded systems shouldn’t fail

Lifecycle: Integration and Testing

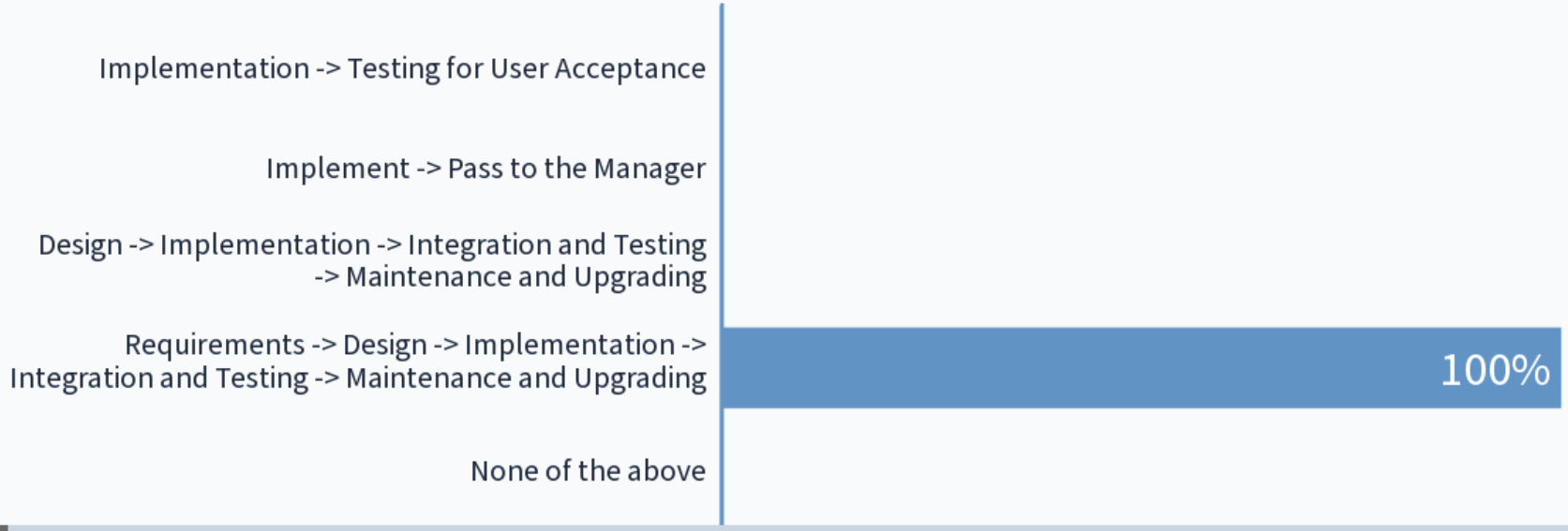
- Objective
  - To get the embedded system ready for user acceptance
- Challenges
  - Time
    - Temporal properties can be one of the hardest properties to prove
  - Deployment
    - Difficult to prove system is ready for deployment

Lifecycle: Maintenance and Upgrading

- Objective
  - To carry out fixes or product enhancement in a managed fashion
- Challenges
  - The connected vs the unconnected
    - Millions of units deployed over many geographical locations
    - May or may not be connected via internet
  - Availability and safety
    - Critical systems cannot be shutdown for maintenance or upgrading
    - No “Windows-style” reboot after patches

Poll locked. Responses not accepted.

Suppose you are working in a company and your manager just asked we need to develop an embedded system product. What would be your approach to it?



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Summarising

- Embedded Systems
  - Definition
  - Examples
  - Usage
  - Features
  - Trends
  - Engineers' desired traits
  - Software development
  - Lifecycle

- Next-> Multi-core Based Next Generation Embedded Systems

Further Questions?