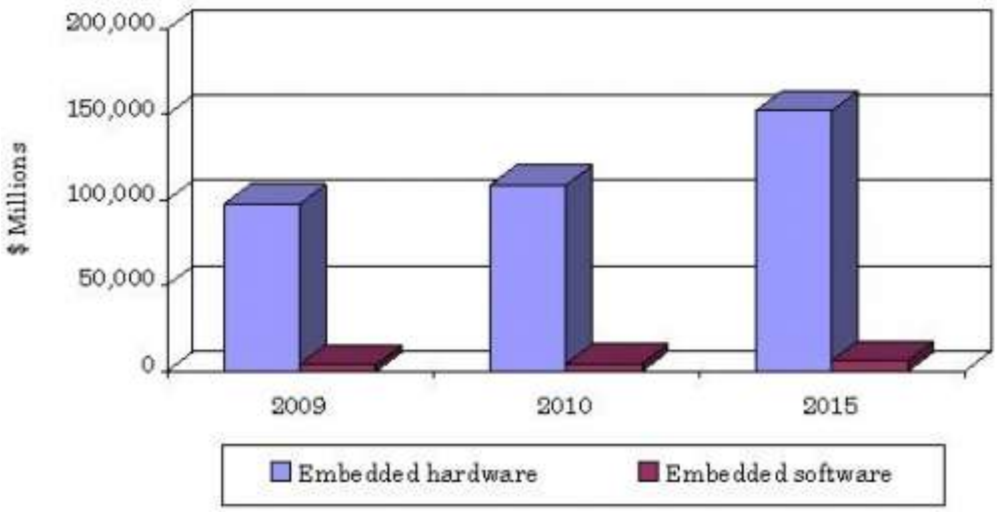
Public Information

"The worldwide market for embedded technology was $101.6 billion in 2009, and was $113 billion in 2010. The market will exhibit steady growth at a compound annual growth rate (CAGR) of 7% over the next 5 years and BCC expects the market to reach $158.6 billion by 2015.

- The embedded hardware segment represents the majority of the overall market with a $108.8 billion share in 2010. BCC expects this market to reach $152.4 billion by 2015 increasing at a CAGR of 7%.
- The software segment accounts for $4.2 billion in revenue in 2010. This segment is expected to grow at a CAGR of 7.8% and will reach $6.1 billion by 2015. "

SUMMARY FIGURE
EMBEDDED TECHNOLOGY MARKET, 2009-2015
($ MILLIONS)



Source: BCC Research

Software Intensive Systems Technology futures – A Viewpoint

Brown-field engineering not Green-field (more System re-engineering, less original system design)

Large/larger scale Systems of Systems

New 'Products' now replacement of components of system, not whole system

System in regular/continuous change

'Enterprise journey' is a continuum – no clean sheet design, living with legacy is norm

Systems of systems increasingly of mixed integrity, safety and security

Provenance of other components of system not necessarily well known/defined

Adaptive systems are the norm

Learning behaviour through 'System Identification' no longer plausible

Systems instrumented to accommodate changed behaviours

Negotiating for optimal working point

Optimising control responses at negotiated working point

Learnt behaviour both locally and from 'external signals'

Externals include 'group' behaviour

Trending information from other products

Monitoring People and Equipment and context aware instruction

Information from everything, in support of decisions
Location and context based services
Ubiquitous interconnect of software systems
- Wireless 'access' technologies, seamless transfer across connections
- Thin client access to huge computational services (e.g. cloud)

Ubiquitous sensing technologies (Personal, Domestic and Industrial)
- Motion processing (MEMS)
- Embedded Vision/Video processing
  - Comparative techniques (should be /as-is)
- Touch free Human interfaces (gesture, context aware, inc voice)
- Position (Geographic and attitude (standing, sitting, walking etc) for context awareness)
- Integrated (personal) health monitoring (Blood pressure, Blood Oxygen levels, Heart Rate)

Augmented reality (video/still overlays, drawings, non-visual spectra)

Social dependence on software (c.f. Mobile phones, credit card transactions)

Increased expectations on software reliability, safety and security
One (wearable) system does it all (communication, data access/recall, vision systems, tools etc)
- Augmented human sensing (extra bandwidth (e.g. visual spectrum) or sensitivity (e.g. noise, smell))
- Augmented non-human sensing (e.g. RF, remote thermal/IR, ELF, Vibration))

Never 'off' systems
Design for Safety is not sufficient
- Security (loss of information, denial of information, misleading information)

Managing transitions of deployed systems and performance becomes new major programme management skill (incremental update for obsolescence or technology insertion)
Software safety walked-up in all lower rigour environments
Software employed at increasingly higher rigour expectations
Architecture and Management of live 'system' updates and configuration (whilst maintaining availability) becomes new skill.

Dealing with Sensory input and Integration – Scalable computational needs

Portability driven in embedded by regular changes of silicon architecture, processor/memory bottlenecks, multi-core and asymmetric cores necessary for intelligent work offload / power
Portability/Scaling become high-value, platform specific knowledge
Portability/Scaling drives new standardisations (interfaces, communications)
Device drivers become 'adaptive', bootstraps identify and 'collect' appropriate software support
Complexity outstrips computational capability (New sensors high computational need)
Complexity outstrips Development Capability ***(Currently 50% are 3 years late, 50% >2x budget)***
Complexity outstrips Affordability ***(Currently 30% projects canned due to cost)***
(Overall complexity growth doubling every 2 years (telecoms), 4 years (automotive), 7 years (aero))
Less static definition – less determinism, clear prioritisation
Frequent Dynamic re-purposing (including remote download/re-programming)
System optimisation (performance) driven out by portability, thru-life adaptability, consumes more compute
Virtualisation support for integrated Multi-integrity systems to ubiquitously house applications

New Business scenarios - Everyone sells their software

- ***(Following assume scenario similar to Audio/Visual market (TV chips and software drivers – note Broadcom ultimately exited from what was once a mainstream arena, due to Asian competition!))***
- ***2008 – Approx 1 Billion Lines of Code generated each year (and increasing) by Software houses and OEMS – with average shelf life of 5 years (decreasing). 6 Bn microprocessors p.a. of which 98% in embedded applications for improved quality of life!***

Good Software developers are costly resource, difficult to attract and retain

Software Resource (like manufacturing) need to be kept productive even when no business-led development.

Must work on High IP-value to be differentiating business expenditure. ‘

Buy’ rather than ‘Make’ is guiding principle.

Resource management suggest Software teams manage bespoke developments

for other (high integrity) opportunities, when not actively working on product

or become trusted partners on external projects for load management.

Software re-use dominates in new form

not just in product or sector

or even within business

but across businesses

Lots of companies sell software as commodity encompassing IP

Independents, Value Added Resellers, Systems Integrators and Consultants

OEMs sell software to other manufacturer (including competitors!) as original cost/risk of Development too high,

initially for common standards implementations,

but progressively more (non-competitive) components.

‘Cost of development’ drives others to secure ‘ready-made’ solutions

even from competitors (especially for areas with no IP like communication protocol standards etc).

Lead developers are able to recoup some (or all) of their outlay if executed to high enough quality.

2nd users get reduced time to market

OEMs get development cost/risk or opportunity costs from not using their engineering workforce.

Customers increasingly unwilling to pay bespoke development costs for what they see as ‘commodity’ item.

Make/Buy pressure driven by quality of sources and timing.

System Integration (Software System Integration) becomes defining role for ‘world-class’ capability recognition

Designing systems for “Ease of use” (hiding complexity) becomes a key market driver

Software Supply Chain management (Being able to spot, and negotiate for, the right system component)

Key for securing right component

Key for securing the right software driver reseller/partner

Both above keys driven by ‘ease of integration’, as this becomes significant development cost.

Validation in high integrity, high-complexity systems, becomes dominant development cost, not implementation.

Software Differentiation is not just on price

- Development Team Experience & Talent
- Architecture
  - Ease of Deployment
  - Quality (Safety, Integrity, Certifiability, Performance etc)
  - Portability (O/S, platform, micro and other hardware devices)
- Support (Bug correction)
- Standards adherence (both Development and Interfacing)
- Test materials and evidence
- Shrink-wrap, or development support
- Open Source solutions

‘Software talent’ becomes a significant capital asset in corporate evaluation

Encompasses all aspects of programmable System realisation through integration

New Industry - Support to Systems Integration

New tools, New techniques, Higher level (more abstract) (inter-box)

Based on analysing and debugging systems (systems debug) of ‘communicating’ systems

Interface snooping, recording, playback and reference to standards (like protocol analysers)

Designing for System integration drives System communication design tools (actors, messages and channels)

System behaviour validated by models of transactional analysis of system from interfaces/messages/channels

Multiple simultaneous interfaces, time correlated

Many concurrent ‘conversational’ threads

Component performance is defined by ‘command’ to ‘output’ or ‘response’ timing

Network analysis, communication bandwidth, protocol needs (e.g. guaranteed delivery) determine distribution of tasks within distributed systems.

Commercial Marine - Example

Challenges: Energy efficiency (vehicle), Emissions; Arctic/Ice operation; LNG, other volatile and/or toxic fuels or cargoes; Deepwater exploration, production and transport; Efficient Construction, Operation and Removal of Offshore facilities; Ocean-based energy; Integrated transport and management (shore-side and ship-board); Economic Logistics of ship, cargo, route and crew?; Piracy (vessel, cargo and crew?); Safety (Deck-handling, maneuvering, man/machine proximity)?; Damage avoidance (Maneuvering, collision, grounding)?;

Brown-field Engineering not Green-field

1. **Business revenue is dominated by upgrades rather than complete new installations /re-fits.**
2. **Opportunities for replacement of competitor equipment and increase of market share**
3. **Detailed specifications of existing equipment largely unavailable or inaccurate (especially if competitors)**
4. **Equipments within system all age at different rates, so rolling changes to system**
5. **Pressure for efficiency demands investment in replacing components with newer efficient technologies**
6. **Optimal working point not a single point 'design' task, needs field capable customisation**
7. **Installation/Commission not necessarily part of contract - so needs automated adaptation**
8. **Need opportunity for re-optimising with other equipment fits to prolong service life**

Monitoring People and Equipment and context aware instruction

1. **Minimum strength, de-skilled crew, large vessel, difficult communication and visibility,**
2. **Harsh human environments, temperature, noise, vibration**
3. **Clear instruction delivery and support over a diverse range of tasks (talking/video manuals)**
4. **Expert support for infrequent/less-practiced tasks (c.f. Computer-based training, delivered JIT)**
5. **Standard automated assistance or remote expert (inc real person!)**
6. **Crew stress relief, Safety support, Automated accident assistance, Cry for help, location id**
7. **Physically large systems with potential for damage/cost (Large masses and energies)**
8. **Intelligent (understands disposition of personnel) fire, smoke, flood control**

Social dependence on software

1. **Reliance on equipment, minimum holding for logistics, software part of argument for safety / training**
2. **Choreography of transitions in service arrangements, equipment hand-off (old to new), commissioning**
3. **Secure, tamper-proof, remote operation and monitoring**
4. **Guaranteed 'full-authority' control (no, or restricted access to, manual options)**
5. **Very high availability expectations**
6. **In-vehicle navigation, Fire, smoke, flood evacuation aids**

Dealing with Sensory input and Integration – Scalable computational needs

1. **Sensing and Vision fidelity**
2. **Augmented human capability (e.g. low light, high noise) signal extraction**
3. **Non-human or high-sensitivity sensory spectra (e.g. infra-red/thermal, biological/olfactory)**
4. **High compute, low power, long service life, extreme (noise, heat, vibration, shock, EMC) environments**
5. **Energy harvesting to supplement local power budget (on person say)**
6. **Input Synthesis; Functionality even in degraded state**
7. **'Universal' equipment, re-purposed by software (c.f. Multimeter, iPad) for logistics, inventory, availability**
8. **Equipment functionality 'growth'**

New Business scenarios - Everyone sells their software

1. **Recovering investment/income stream for 'standards' (e.g. Communication protocols)**
2. **Easier marketing/integration of equipment (complete with xx-standard s/w drivers!)**
3. **Faster development internally (c.f. ABB, Siemens S7) – cuts cost, risk and time**
4. **Faster deployment and commissioning (of equipment to supported standards)**
5. **Provenance of re-used equipment**
6. **Ease of use and 'Family' feel to equipment operation (look and feel), transferable operating skill**
7. **'Simplicity', less training, more demand from end-user, High-value (& profile) HMI, low user-failure risks**
8. **Brand association for reliability and simplicity ('Trusted to deliver excellence')**
9. **Value by association (trusted to supply Rolls-Royce) for software/equipment (c.f. mechanical)**

New Industry - Support to Systems Integration

1. **Rolling programmes of updates managed by System suppliers (including competitors equipment)**
2. **Ability to diagnose, isolate and repair/replace systems rapidly**
3. **Confidence in coherent design solutions**
4. **Coherent safety arguments, safe operation, design margin for ship designers and operators**
5. **Supporting classification societies needs whilst assuring system functionality**
6. **Clear design-supported operational capability statements (especially failure scenarios)**