KTH Informations- och kommunikationsteknik

IS1300 Inbyggda system

Bengt Molin

Tel: 08-790 4448

E-post: bengtm@kth.se

EKT – Komponenter och kretsar

ICT-skolan, Informations- och kommunikationsteknik

Kista, Electrum, hiss C, plan 4

Föreläsning 1-2 – Inbyggda system

- Kursinformation
- Examination
- Labuppgifter – egen laptop
- Vad är ett inbyggt system?
- Ett exempel
- Vilka kunskaper behövs för att utveckla ett inbyggt system?
- Skriva rapport

Kursinformation

- Examinator, kursansvarig Bengt Molin
- Kursregistrering
- Betygsättning
  - Lab P/F (4,5 hp) Konstruktionsuppgifter
  - Tenta A-F (3 hp) Skriftlig tentamen
- Kursmål, se kurswebbsida, kurs-PM
- Studentrepresentanter kursnämnd: **? ?**
- Problem i schema ?

Examination

LAB1 (4,5 hp) Betyg P/F **TEN1 (3 hp)** Betyg A-F

Inledande uppgift pingpong **Villkor för att få fortsätta kursen**

Kursbetyget kommer att sättas enligt följande poängsystem (VT15)

Större uppgift	
Planering, arkitektur, struktur, testning	0, 1, 2
Genomförande, komplexitet	0, 1, 2, 3
Rapport, dokumentation	0, 1, 2
Realtidsuppgift	0, 1, 2, 3
Skriftlig tentamen	0, 1, 2

Slutbetyg

A	**12**
B	**10-11**
C	**8-9**
D	**6-7**
E	**5**

Obligatorisk närvaro vid
Seminarier (2 st)
Laborationer (6 st + redov)

Kursen

Föreläsningar

Tenta

Inledande uppgift pingpong

Större uppgift Inbyggda system

Realtidsoperativsystem

Bra exempel; e-athletics

- Vann Embedded Award 2013
  - Kategori mikro/nano
- Uppmärksammad i press
  - Lokalt
    - VLT
    - Provins
  - Fackpress
    - Uppfinnaren och Konstruktören
    - Process Nordic

FRÅN SLÄGGMEDALJÖR TILL ENTREPRENÖR

KJELL BYSTEDT TAR ETT VARV TILL

Den tidigare svenska mästaren i slägga, västeråsaren Kjell Bystedt, dyker återigen upp i både rubriker och kastring. Inte nog med att han är tränare åt nuvarande svenska släggmästaren – Mattias Jons – han arbetar dessutom för att föra in elektronik och inbyggda system i sporten genom sitt e-athletics.

Text: Björn Stenvall · Foto: Firma Kjell Bylund, Svensk Friidrott

Kjell Bystedt under sin glansdagar som aktiv svenska mästaren i släggkastning Mattias Jons.

-athletics representerar en mycket avancerad teknik i ett sammanhang där tekniken ger ganade

INNOVATION |

Swedish Embedded Award

Swedish Embedded Award delas sedan 2006 ut av Branschorganisationen Svensk Elektronik. Priset som är branschens stora innovationspris delas ut i syfte att visa på de fantastiska möjligheter som öppnar sig med embedded technology. Priset delas ut i de tre olika kategorierna: **Enterprise**, för svenska företag, **Student**, för studenter på Sveriges högskolor, och **Micro/Nano**, för studenter och företag.

Årets vinnare i klassen "Enterprise Category" blev Dasa Control Systems AB med bidraget Forester, som är ett så kallat apterings- och styrsystem till skogsmaskiner och då främst till skördare. Systemet är ett avancerat, men lättanvänt, grafiskt gränssnitt där föraren av skogsmaskinen kan styra avverkningen för att få optimalt utbyte till sågverk eller massaindustri. Syste-

MOT NYA REKORD. *Svenska mästaren i släggkastning Mattias Jons med den nya mätenheten i släggan. Sensorn syns som en genomskinlig punkt på kulan som Jons håller i handen.*

Mätbar mikroelektronik – kul idé blir nästa stora sak inom friidrotten?

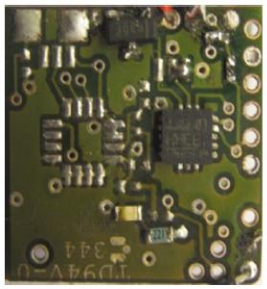
sultat har man under de senaste 30-40 åren använt ett batteri av olika åtgärder, från anpassning av redskap till nya material, kosthållning, utstyrsel och annat. Men det som är relativt nytt är att elektroniken börjar komma in i idrottsredskap.

BRANSCHORGANISATIONEN SVENSK ELEKTRONIK

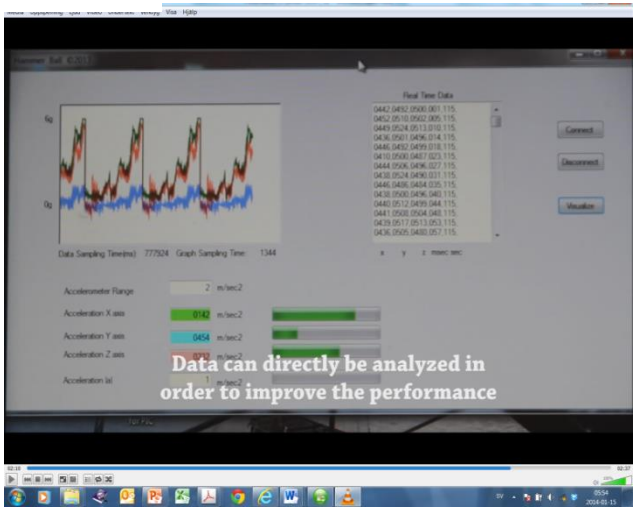
Kommersialisering

- Från studentprojekt till företag

Projektägare med samarbetspartner: Kjell Bystedt, Athar Nadeem Ahmed och Stefan Söderlund.



- Nya tillämpningar



- I samarbete med SE-medlemmar och i nätverket runt SE

BRANSCHORGANISATIONEN SVENSK ELEKTRONIK

Embeddedpriset.nu

Definition of Embedded System

- Special-purpose computer system designed to perform a limited / dedicated number of functions often in real-time
- Any device that includes a programmable computer but is not itself a general-purpose computer.

- Typically consists of
  - Microprocessor (MPU) / Microcontroller (MCU)
  - Memory
  - I/O ports
  - Sensors
  - Communication

Characteristics of Embedded Systems

- Designed for specific purpose
- Interacts with environment
- Real-time operation
- Controlled by embedded computer
- Low manufacturing cost
- Low power
- Designed to tight deadlines

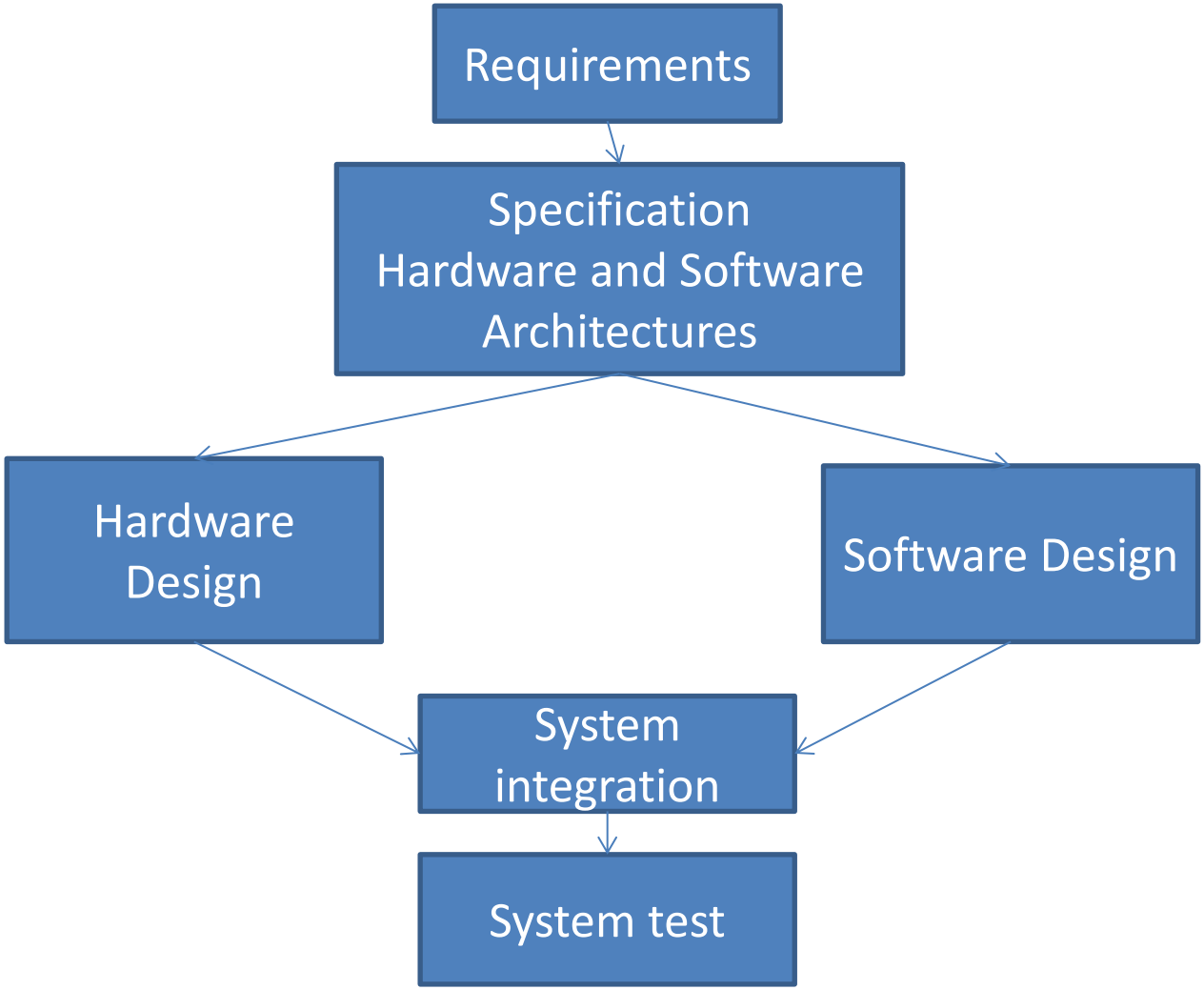
Design Challenges

- Price, low cost
- Light weight
- Reliability
- Low power
- Portable
- Complexity

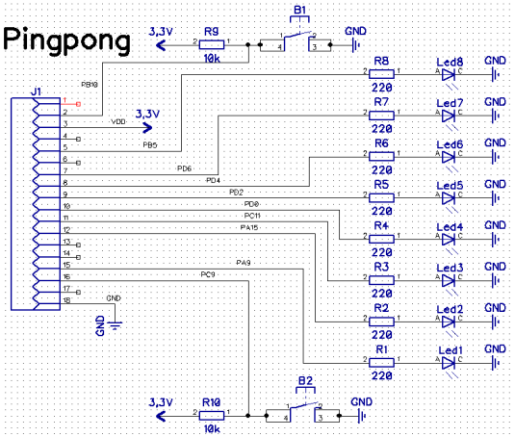
Challenges in embedded design

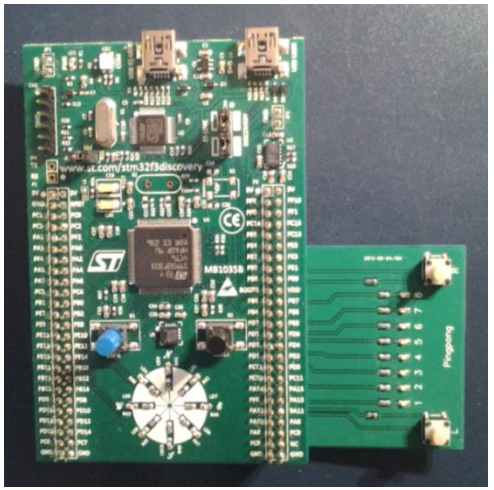
- How much hardware do we need?
- Choice of microcontroller?
- Choice of development platform?
- How do we meet our deadlines?
- Faster hardware or better software?
- How do we minimize power?
- Does it really work? How to test the system?

Design process



Presentation Uppgift Pingpong





Det finns två spelare, L och R (Left och Right). Den som servar trycker på sin knapp. Då tänds lysdioder, en i taget, för att animera att en boll rör sig till motspelaren. En boll returneras om knappen trycks ned samtidigt som yttersta dioden är tänd. Om man trycker för tidigt eller för sent är det en tappad boll och motspelaren får poäng. En missad boll markeras genom att alla lysdioder blinkar till lite kort, varefter poängställningen visas. Poängställningen efter en vunnen och tappad poäng visas ett kort ögonblick genom att tända upp så många lysdioder på respektive sida som motsvarar antal poäng. Spelarna turas om att serva. Vid start kan vem som helst serva, men det är den andra spelaren som servar nästa gång. Om till exempel vänster spelare servar första gången är det höger spelare som servar nästa gång oberoende av vem som tappat poäng. Den som först kommer till fyra vunna poäng vinner och spelet avstannar. Den tid varje lysdiod är tänd skall minska allteftersom hur länge en boll varit i spel. På så sätt ökar ”bollhastigheten” och det blir allt svårare att hinna returnera bollen.

Hur löser vi detta problem

Du skall lösa denna uppgift fram till fredag nästa vecka!

Villkor för att få fortsätta kursen!

Hur ska vi testa programmet?

- Traditionellt DLP = Debug Later Programming
- TDD = Test Driven Development
  1. Add a small test
  2. Run all the tests and see how the new one fail
  3. Make the small changes to pass the test
  4. Run all the tests and see the new one pass
  5. Refactor to remove duplication and improve expressiveness

Refactor: Changing structure without changing its behavior. Cleaning!

Bok: Grenning, Test-Driven Development for Embedded C

Ref: http://en.wikipedia.org/wiki/Test-driven_development

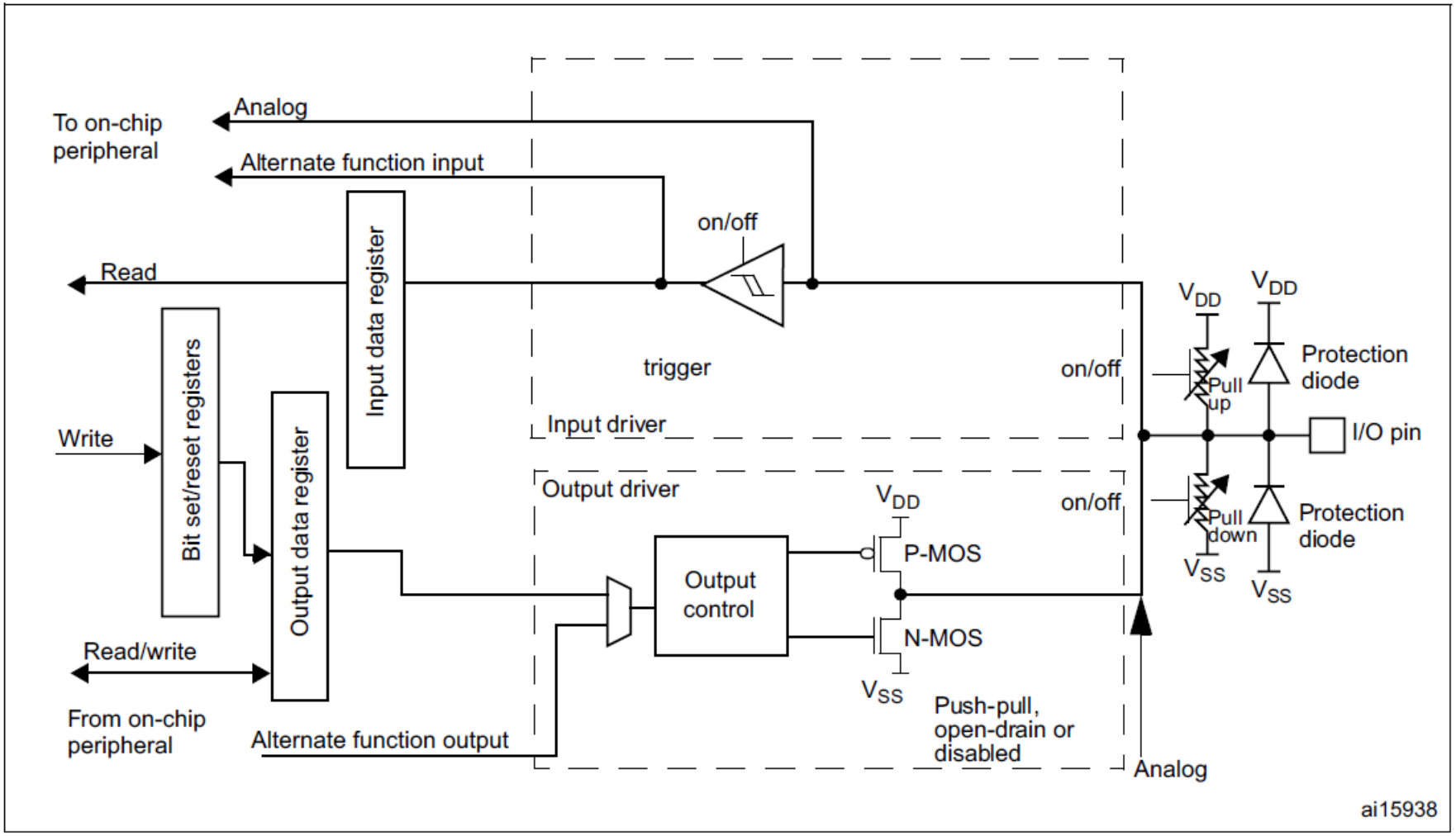
Vårt målsystem STM32F3 Discovery

- Microcontroller STM32F303VCT6
- Dokumentation du behöver ha tillgång till (hämtas från st.com)
  - Manual Discoverykortet UM1570
  - Datablad STM32F303x
  - Referensmanualen RM0316
  - Manual Peripheral Library UM1581

Exempel på initiering av GPIO

- GPIO = General Purpose Input Output, vanliga digitala in- och utgångar
- Initiering av klocka; Clock Tree
- Initiering av port
- Control register
- Status register
- Data register

Digitala In- och utgångar



STM32CubeMX

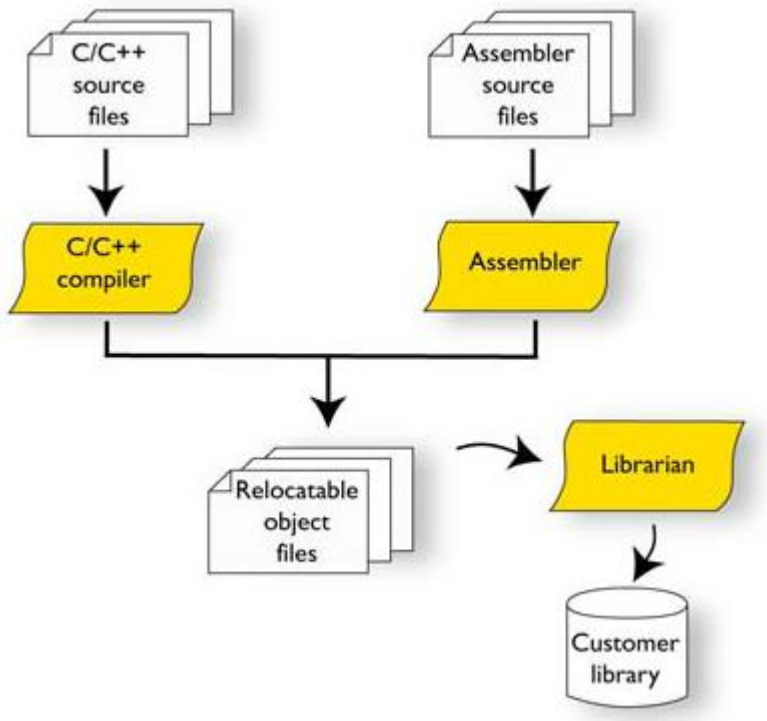
- Hjälp med initieringskod kan erhållas med programmet STM32CubeMX
- Programmet kan hämtas från STMicroelectronics webb www.st.com
- Länk till STM32CubeMX (UM1718)
- Demo

Kompilering och länkning

- För att få ett körbart program måste programmet
  - Kompileras
  - Länkas
  - Läggas in i minnet på rätt ställe
  - Initieras

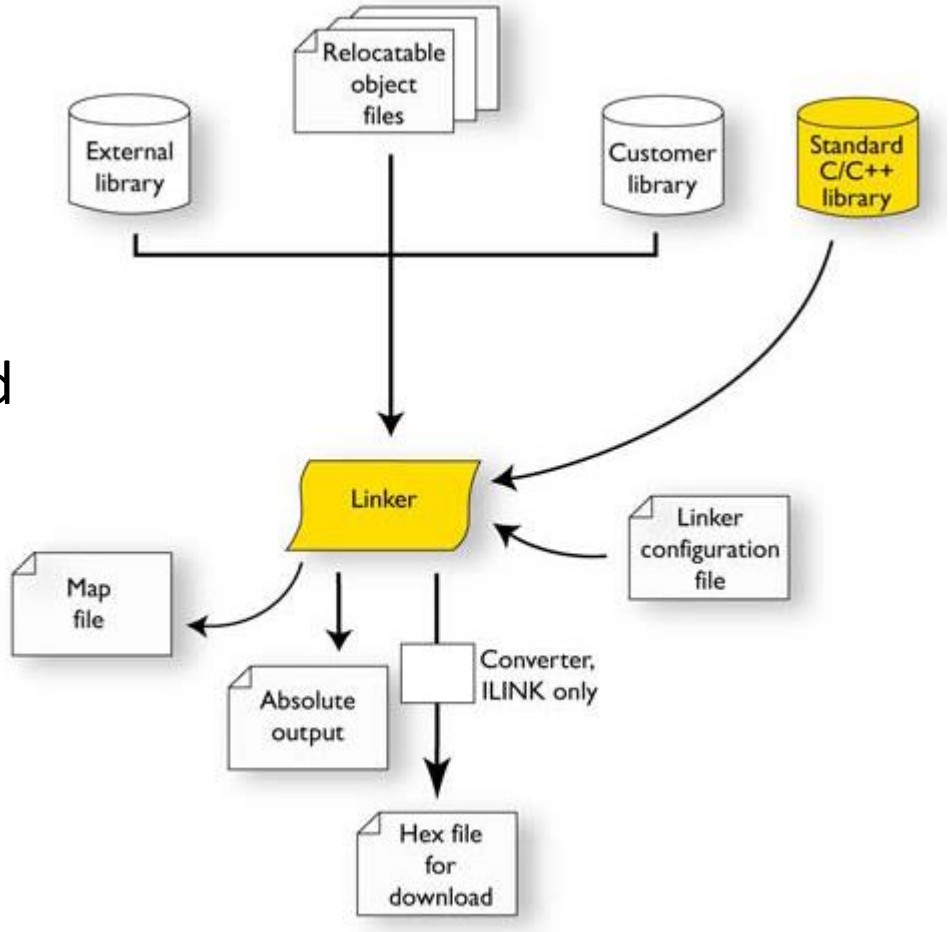
Compiler and assembler

From source code to object code

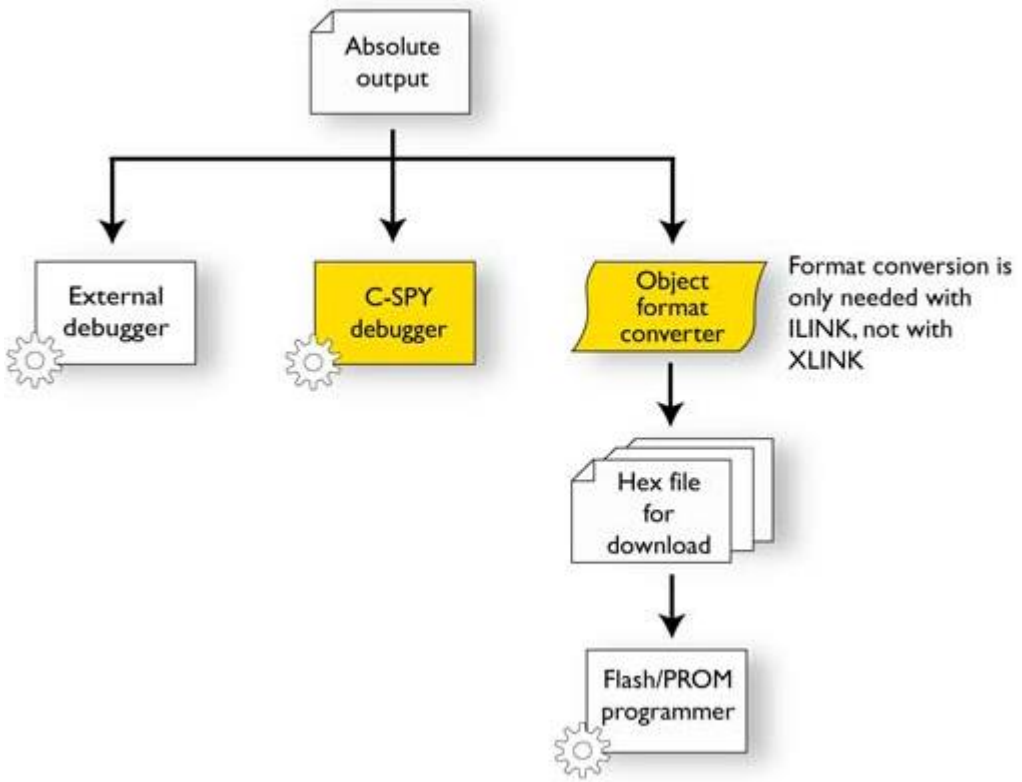


Linker

Linking object files and library files to an executable HEX file for download



Linker output



Initialization phase after reset

Initialization is executed when an application is started (the CPU is reset) but before the main function is entered

● Hardware initialization, for example initializing the stack pointer
The hardware initialization is typically performed by the system startup code Cstartup.

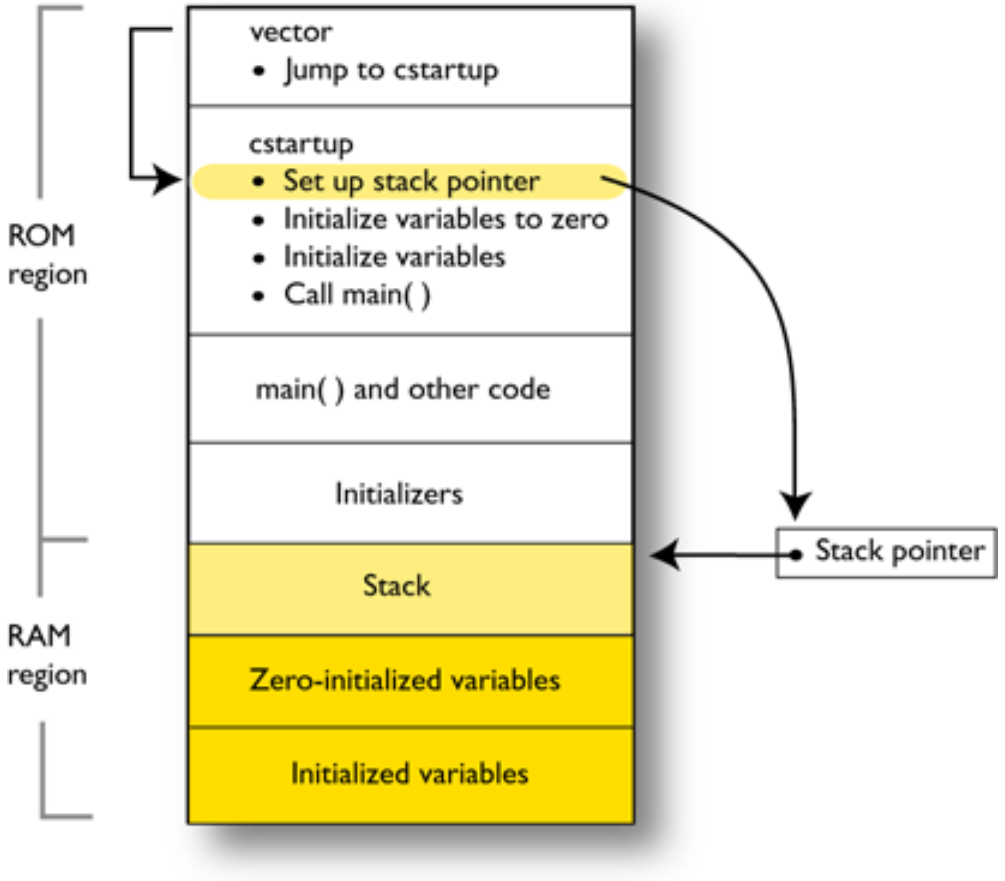
● Software C/C++ system initialization
Typically, this includes making sure that every global (statically linked) C/C++ object receives its proper initialization value before the main function is called.

● Application initialization
For a bare-bone application, it can include setting up various interrupts, initializing communication, initializing devices, etc.

For a ROM/flash-based system, constants and functions are already placed in ROM. All symbols placed in RAM must be initialized before the main function is called. The linker has already divided the available RAM into different areas for variables, stack, heap, etc.

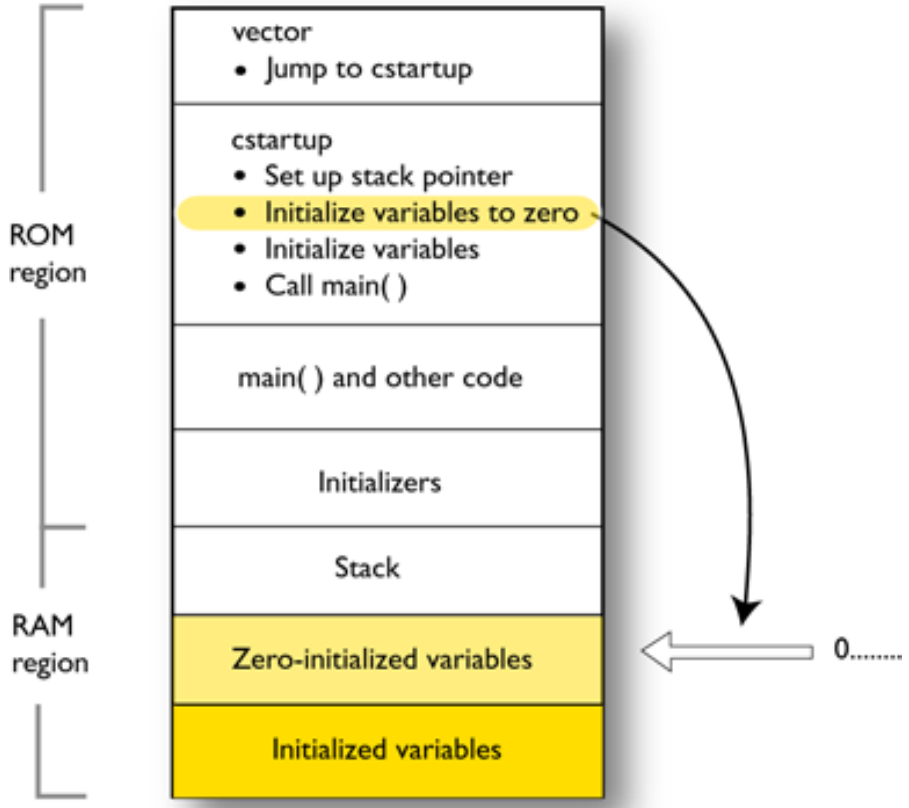
1 Initialize stack pointer

When an application is started, the system startup code first performs Hardware initialization, such as initialization of the stack pointer to point at either the start or the end—depending on your microcontroller—of the predefined stack area.



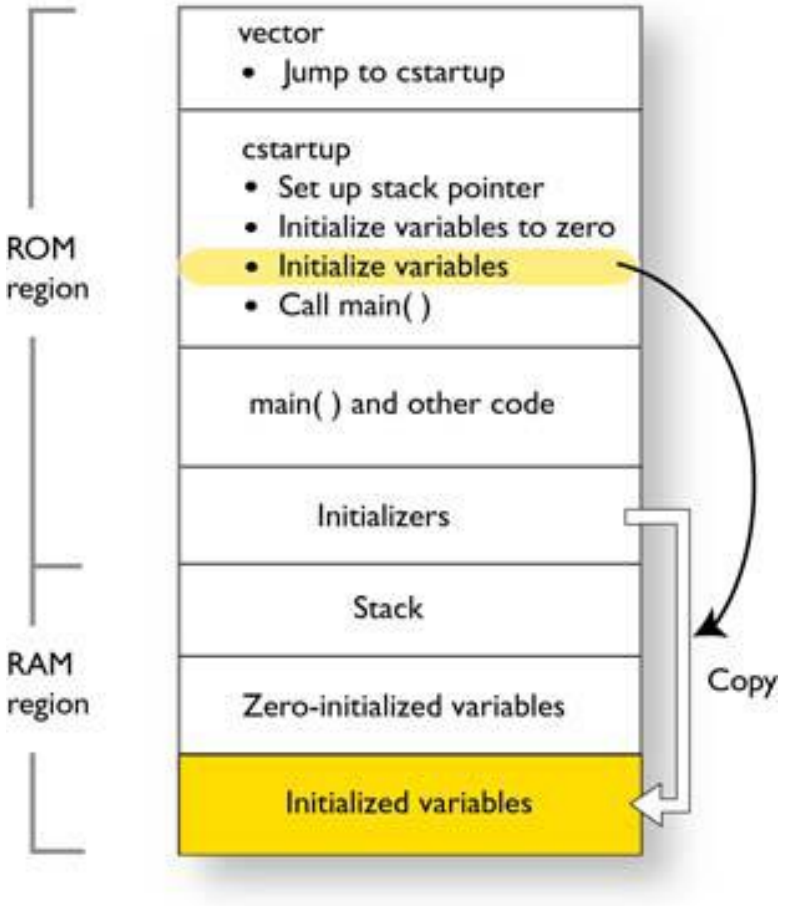
2 Zero-initialized variables

Then, memories that should be zero-initialized are cleared, in other words, filled with zeros. Typically, this is data referred to as zero-initialized data; variables declared as,
for example, int i = 0;



3 Initialize data

For *initialized data*, data declared with a non-zero value, like int i = 6;, the initializers are copied from ROM to RAM.



4 Finally, main is called

Finally, the main function is called.