

Maskinorienterad Programmering

EDA488 för Z

EDA483 för E

DIT151 för GU

LP4, 2023

Kursöversikt:

- Genomförande,
- Litteratur/läromedel,
- Syften och målsättningar

Kursens hemsida (Canvas)

▼ Lektioner
 Kursintroduktion 22 mar
 Lektion 1 - Introduktion till Maskinorienterad programmering 24 mar
 Lektion 2 - Korsutveckling för ARM/Thumb del 1 24 mar
 Lektion 3 - Korsutveckling för ARM/Thumb - del 2 29 mar
 Lektion 4 - Fält, pekare och portar 31 mar
 Lektion 5 - Pekare 5 apr
 Lektion 6 - Digital IO 7 apr
 Lektion 7 - Synkronisering 7 apr
▼ Laborationer
▼ Handledning och diskussionsforum
▼ Kursmaterial
 Dokument på elektronisk form
 Exempelsamling med lösningar
 Programvaror som används i kursen
 Tidigare tentamina

Information om lektioner, laborationer, handledningstillfällen och kursmaterial.

Flikar öppnas efter hand under
kursens gång

Kursnämndsrepresentanter:

EDA488:

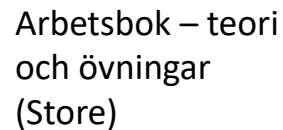
EDA483:

DIT151:

Någon lämplig bok om 'C' – exempelvis:

- "The C-programming language", Kernighan/Ritchie

Eller: "Snabb introduktion till C" och resurser på nätet.




CHALMERS

Kursutveckling för ARM/Thumb – del 1

Ur innehållet:
 Programutveckling i C och assembler
 Programutvecklingsmiljö för GCC/ARM
 Korskompilering

Maskinorienterad programmering **odgenerering för ARM**



Introduktion till Maskinorienterad programmering

Målsättningar:

- ✓ bli bekant med utvecklingsmiljön
- ✓ Kunna redigera, kompilera och testa ett enkelt C-program "Hello world" på egen hand...

Ur innehållet:

- ✓ Programutveckling i C och assembler
- ✓ Programutvecklingsmiljö
- ✓ Så översätts ett C-program
- ✓ Introduktion till C
- ✓ Variabeldeklarationer
- ✓ Heltalstyper
- ✓ Funktioner
- ✓ Textsträngar
- ✓ Åtkomlighet och synlighet






len



Maskinorienterad programmering



Learn C Programming

[TUTORIALS](#)[EXAMPLES](#)[REFERENCES](#)[ONLINE COMPILER](#)

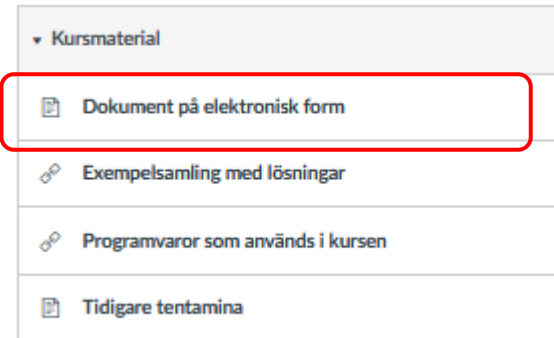
C is a powerful general-purpose programming language. It can be used to develop software like operating systems, databases, compilers, and so on. C programming is an excellent language to learn to program for beginners.

Our C tutorials will guide you to learn C programming one step at a time.



The Definitive Guide to ARM® Cortex®-M3 and Cortex-M4 Processors, Third Edition, Joseph Yiu, ARM Ltd, Cambridge, UK.

Resurser – direkt från kurshemsidan



Grundläggande material för självverksamhet:

- Snabb introduktion till C, kompendium [\[PDF ↗\]](#)
- Inledande hemuppgift 1, "Fibonacci", [\[PDF ↗\]](#)
- Inledande hemuppgift 2, "Goldbach hypotes", [\[PDF ↗\]](#)
- ETERM8 och MD407 inledande övning 1, [\[PDF ↗\]](#)
- GDB och SimServer med MD407, inledande övning 2, [\[PDF ↗\]](#)
- GCC och Simserver för MD407, inledande övning 3, [\[PDF ↗\]](#)
- SimServer/IO-simulator referenshandbok [\[PDF ↗\]](#)
- Laborationsuppgifter för simulatoranvändning [\[PDF ↗\]](#)
- Quick Guide MOP, [\[PDF ↗\]](#)

Fördjupning - Handböcker och datablad - laborationsdator och laborationskort:

- GCC 9.5 [\[PDF ↗\]](#), [\[HTML ↗\]](#)
- GCC 9.5, preprocessor [\[PDF ↗\]](#), [\[HTML ↗\]](#)
- MD407 beskrivning [\[PDF ↗\]](#)
- STM32F4xx Cortex M4 programming manual [\[PDF ↗\]](#)
- STM32F407 datasheet [\[PDF ↗\]](#)
- STM32F407 reference manual [\[PDF ↗\]](#)
- Lysdiodstapel ("bargraph") [\[PDF ↗\]](#)
- Binär till 7-segment [\[PDF ↗\]](#)
- DIL-switch [\[PDF ↗\]](#)
- LCD-modul [\[PDF ↗\]](#)
- Hexadecimal till 7-segment [\[PDF ↗\]](#)
- Keypad [\[PDF ↗\]](#)
- IRQ Flip-Flop [\[PDF ↗\]](#)
- Datablad LCD - ASCII [\[PDF ↗\]](#)
- Datablad LCD - Grafisk [\[PDF ↗\]](#)

Aktuell lärobok är utgåva 6.3.

För dig som använder förra årets utgåva (6.2), är de stora ändringarna att tidigare kapitel 1 och 3 har utgått och ersatts av "inledande övning.." (se ovan).

Kapitel 2 (kapitel 1 i utgåva 6.3) har bearbetats väsentligt, en rad nya övningsuppgifter med lösningsförslag har tillkommit.

[Errata](#) [↗](#) till utgåva 6.3.

Resurser – direkt från kurshemsidan

▼ Kursmaterial

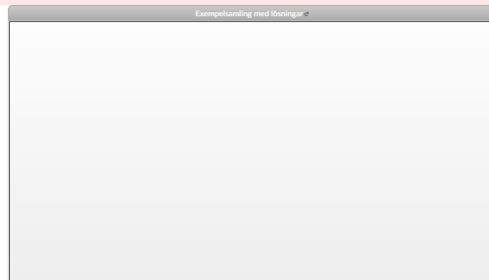
 Dokument på elektronisk form

 Exempelsamling med lösningar

 Programvaror som används i kursen

 Tidigare tentamina

De fönster öppna ett oöskert innehåll från en säker plats (canvas). Visa webbläsare kan förhindra att innehållet bedöms.



Maskinorienterad Programmering Exempelsamling

- [Inledande uppgifter i C](#)
- [Bitmanipulering i C](#)
- [Strängar](#)

Kodgenerering för ARM/Thumb

- [Datadeklarerationer och tilldelningar](#)
- [Typkonverteringar](#)
- [Uttrycksavslutning](#)
- [Subrutiner-parametrar,returvärden och lokala variabler](#)
- [Fält och sammansatta typer](#)
- [Pekare ,pekararitmetik](#)

Periferikretsar

- [GPIO Initiering/programmering digital I/O](#)
- [SYSTICK programmering av enkel räknarkrets](#)

Undantag och avbrott

- [Externa avbrott](#)

Exempelsamling i Maskinorienterad programmering

Uppgifterna i denna exempelsamling anknyter till lektioner i kursen. Teori och praktiska exempel varvas. Åtskilliga exempel är exekverbara och kräver då att du har tillgång till programutvecklingssystemen (Codelite, Eterm8 och SimServer).

Version 2022-04-02. Observera att exempelsamlingen fortlöpande kompletteras med nya uppgifter.

Resurser – direkt från kurshemsidan

▼ Kursmaterial

Dokument på elektronisk form

Exempelsamling med lösningar

Programvaror som används i kursen

Tidigare tentamina

Du försöker öppna ett osäkert innehåll från en säker plats (canvas). Vissa webbbläsare kan förhindra att innehåll laddas.

Programvaror som används i kursen

Programvaror

Datum	Beskrivning			
2023.01.20	ETERM8 och SimServer VERSION=1.04 (0.994) OBS: SimServer 0.993 (MacOS och Linux) kraschade. Denna distribution är uppdaterad med SimServer 0.994 Programmen har paketerats tillsammans för att förenkla handhavandet.	eterm8-VER-setup.exe	eterm8_VER_amd64.deb eterm8_VER_amd64.deb.md5 eterm8_VER.tar.gz eterm8_VER.tar.gz.md5	eterm8+simserver-VER.zip
2022.06.21	CodeLite GCC och GCC för ARM Här finns det skillnader beroende på din plattform. För Windows har vi förberett ett komplett paket med CodeLite, GCC för Windows och GCC för ARM. För Linux sker installationen i flera steg, GCC för Linux är vanligtvis redan installerad. GCC för ARM och konfigurationsfiler för korskompilering laddar du hem härifrån. CodeLite installerar du direkt från dess hemsida. För MacOS gäller att du först måste installera utvecklingspaket ("Xcode Tools") om du inte tidigare har gjort detta. Utöver detta behöver du det paket med CodeLite, GCC för ARM och konfigurationsfiler för korskompilering som du laddar hem härifrån. Alla instruktioner du behöver hittar du längre ned.	codelite-amd64-16.0.0-cse.exe	GCC 9.2.1 för ARM Konfiguration GCC Mallar för projekt	codelite-cross(14.0.5).zip
2020.12.12	Exempelsamling Exempelsamlingen är web-baserad (öppnas i ny tab här) och fylls på efter hand. Vi håller den "bakåt kompatibel" så att inte referenser till befintliga uppgifter ändras då nya läggs till. Behöver du exempelsamlingen "off-line" kan du också ladda hem ett komprimerat arkiv. Efter att ha packat upp arkivet startar du filen "index.html" i din web-läsare.			

+ Installation ETERM8 och SimServer under MS Windows

+ Installation av ETERM8 och SimServer under Linux

+ Installation av ETERM8 och SimServer under MacOS

+ Installation av CodeLite, GCC och GCC för ARM under MS Windows

+ Installation av CodeLite och GCC för ARM under Linux

+ Installation av CodeLite och GCC för ARM under MacOS

Resurser – direkt från kurshemsidan

▼ Kursmaterial
 Dokument på elektronisk form
 Exempelsamling med lösningar
 Programvaror som används i kursen
 Tidigare tentamina

Tidigare salstentamina:

- [EDA488 2022-08-19](#) ➞
- [EDA483 2022-08-19](#) ➞
- [EDA488 2022-06-02](#) ➞
- [EDA483 2022-06-02](#) ➞
- [2022-03-14](#) ➞

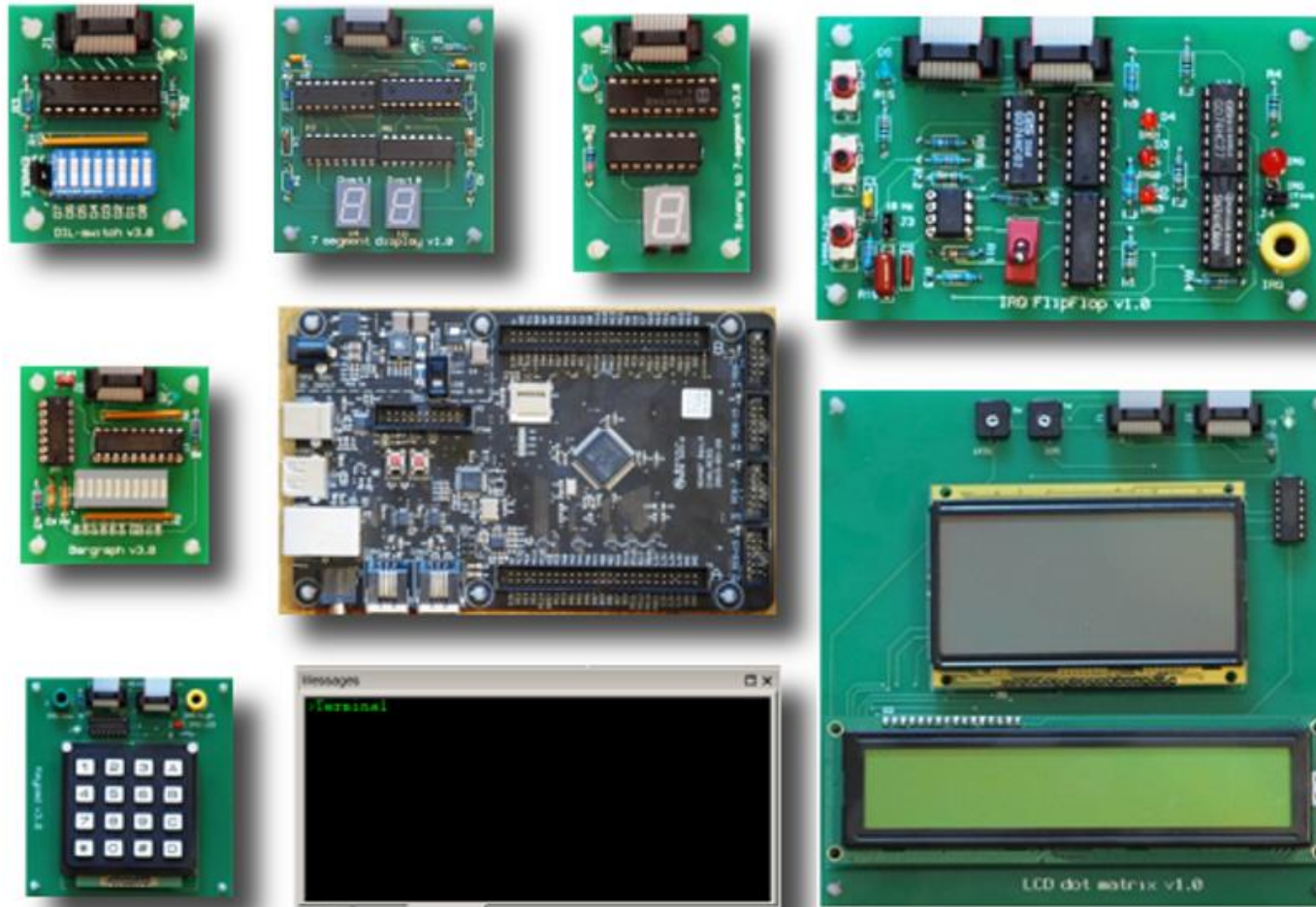
LP3 2020-Augusti 2021 , dessa tentamina har getts som hemtentamen:

- [2021-08-20](#) ➞
- [2021-06-04](#) ➞
- [2021-03-15](#) ➞
- [2020-08-21](#) ➞
- [2020-06-05](#) ➞
- [2020-03-16](#) ➞

Tidigare (före LP3 2020) salstentamina:

- [2019-08-23](#) ➞
- [2019-06-08](#) ➞
- [2019-03-18](#) ➞
- [2018-06-02](#) ➞
- [2018-04-04](#) ➞
- [2018-03-12](#) ➞
- [2018-01-09](#) ➞

Laborationssystem



Kursens syften och målsättningar

□ Kursens syften är

- att vara en introduktion till konstruktion av små inbyggda system och
- att ge en förståelse för hur imperativa styrstrukturer översätts till assembler
- att ge en förståelse för de svårigheter som uppstår vid programmering av händelsestyrda system med flera indatakällor.

□ Centrala målsättningar är att kunna:

- skriva enkla C-program med användande av programspråkets datatyper och styrstrukturer
- beskriva motsvarigheten i assembler till typiska programstrukturer i C.
- utnyttja de i kursen använda verktygen för programutveckling på ett adekvat sätt
- medverka vid konstruktion och programmering av enkla inbyggda system med givna komponenter
- konstruera system innefattande olika typer av undantag (interna undantag, avbrott, återstart)
- beskriva och exemplifiera några olika typer av digitala kringkomponenter och deras användning.

Inför Lektion 1 (Onsdag) och 2 (Torsdag)

- ✓ Installera programvara ETERM8/SimServer och CodeLite
- ✓ Se videoklippen inför lektion 1 och 2

Efter lektionerna:

- ✓ Se anvisningarna kring lektionen
- ✓ Gör den inledande uppgiften "Fibonacci"
- ✓ Om du tycker det behövs, se "Snabb introduktion till C"
- ✓ Gör uppgiften "Goldbach hypotes".

En av institutionens labsalar (ED-4225) är öppen Tisdag 21/3 (10-12 och 13-17) för dom som vill. Här kan man få hjälp med installation av programvara, titta på labutrustningen etc.

Förberedelser inför lektionen:

- Om detta är din första riktiga kontakt med "C" kan du behöva ytterligare litteratur. Du kan exempelvis börja med denna "Snabbintroduktion till C" [\[PDF e\]](#) här finns också länkar till bra resurser på nätet.
- Titta på de följande videoklippen.

Efter lektionen:

Det är nu viktigt att du får i gång den grundläggande utvecklingsmiljön så att du kan arbeta hemifrån när du vill. Du måste också kunna följa, repetera övningsexempel och lösa egna uppgifter. Detta kräver i de flesta fall att du har tillgång till en fungerande utvecklingsmiljö. Kontrollera installationen genom att göra uppgiften "Fibonacci" [\[PDF e\]](#). Därefter kan du ta dig an "Goldbach hypotes" [\[PDF e\]](#).

I exempelsamlingen kan du studera "Inledande uppgifter i C e" 3.1.1-3.1.11 och "Bitmanipulering i C e" 3.2.1-3.2.6