

Programmering av inbyggda system 2013/2014

Kursintroduktion

Roger Johansson

Viktor Kämpe

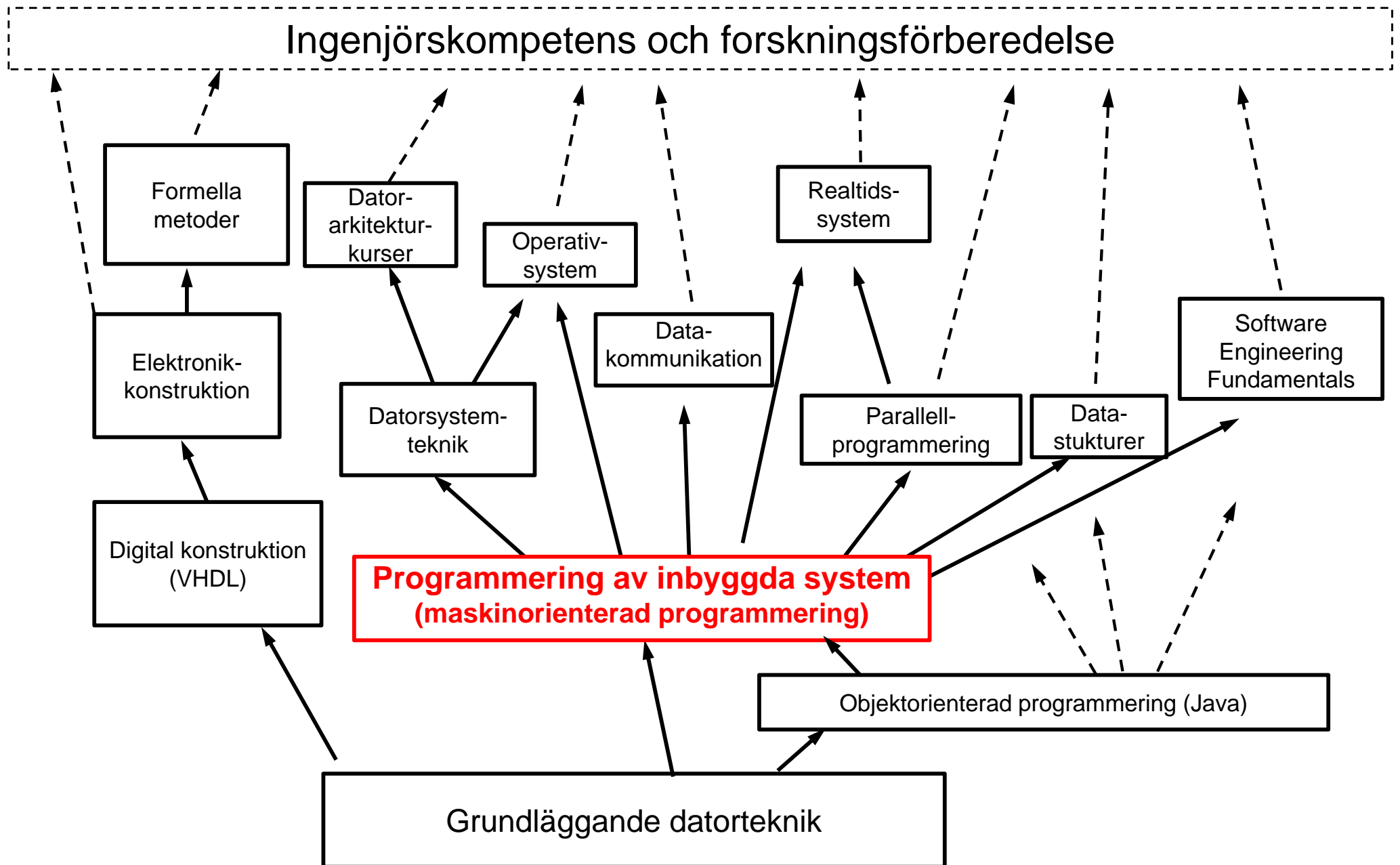
Ur innehållet:

Syften, målsättningar, kurslitteratur och genomförande

Översikt av laborationer

Syften och målsättningar

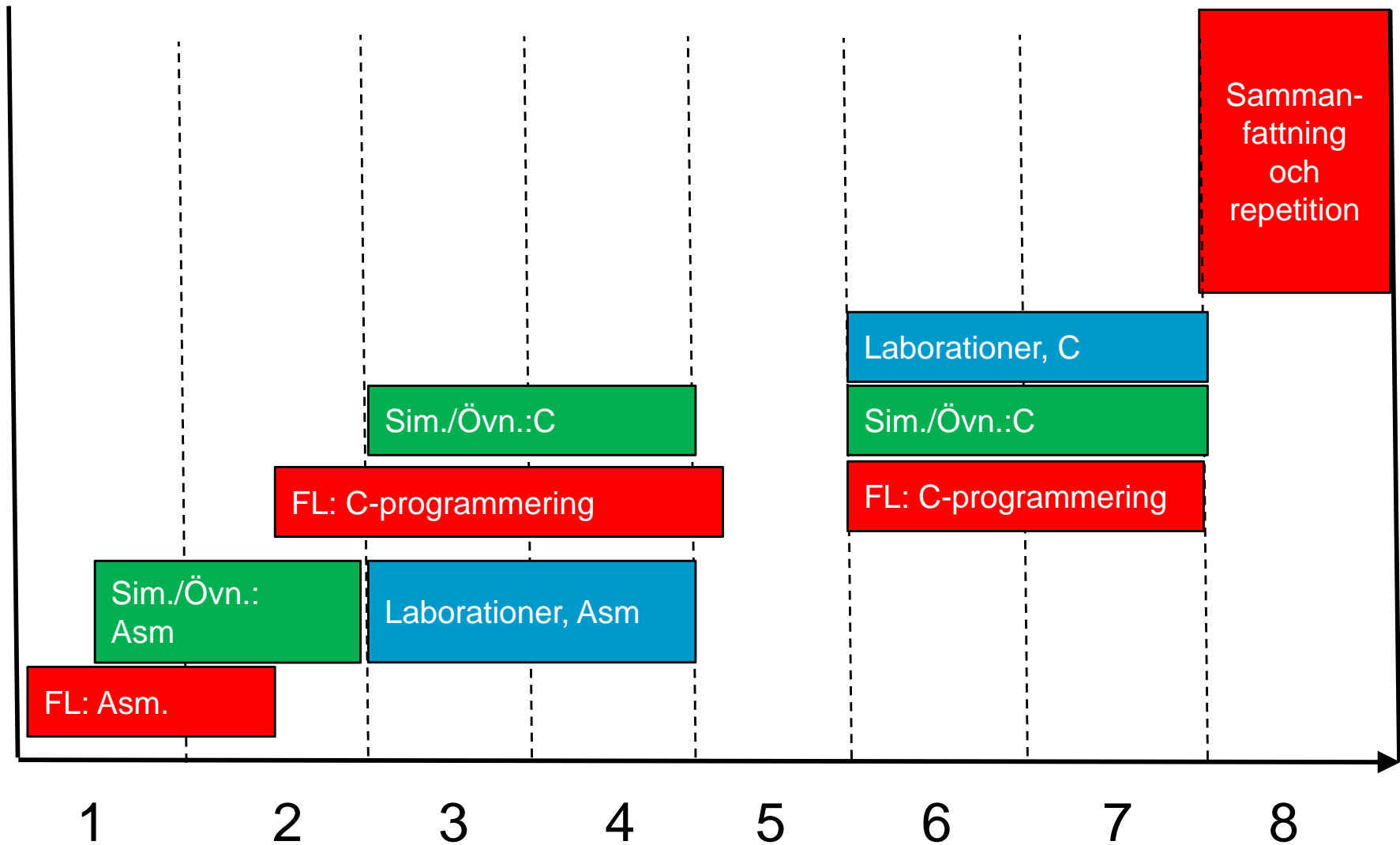
- ❑ Kursens syften är
 - ❑ att vara en introduktion till konstruktion av små inbyggda system och
 - ❑ att ge en förståelse för hur imperativa styrstrukturer översätts till assembler
 - ❑ att ge en förståelse för de svårigheter som uppstår vid programmering av händelsestyrda system med flera indatakällor.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



Kurslitteratur

- Vägen till C (DC)
- Arbetsbok för MC12 (DC)
- Schema och veckoplanering,
se länk från kursens hemsida (PingPong)
- Laborations-PM, delas ut under lektioner och finns även på
elektronisk form (via kursens hemsida).
- Övrigt material, på elektronisk form kan också hämta via kursens
hemsida.

Genomförande



Vecka

1

2

3

4

5

6

7

8

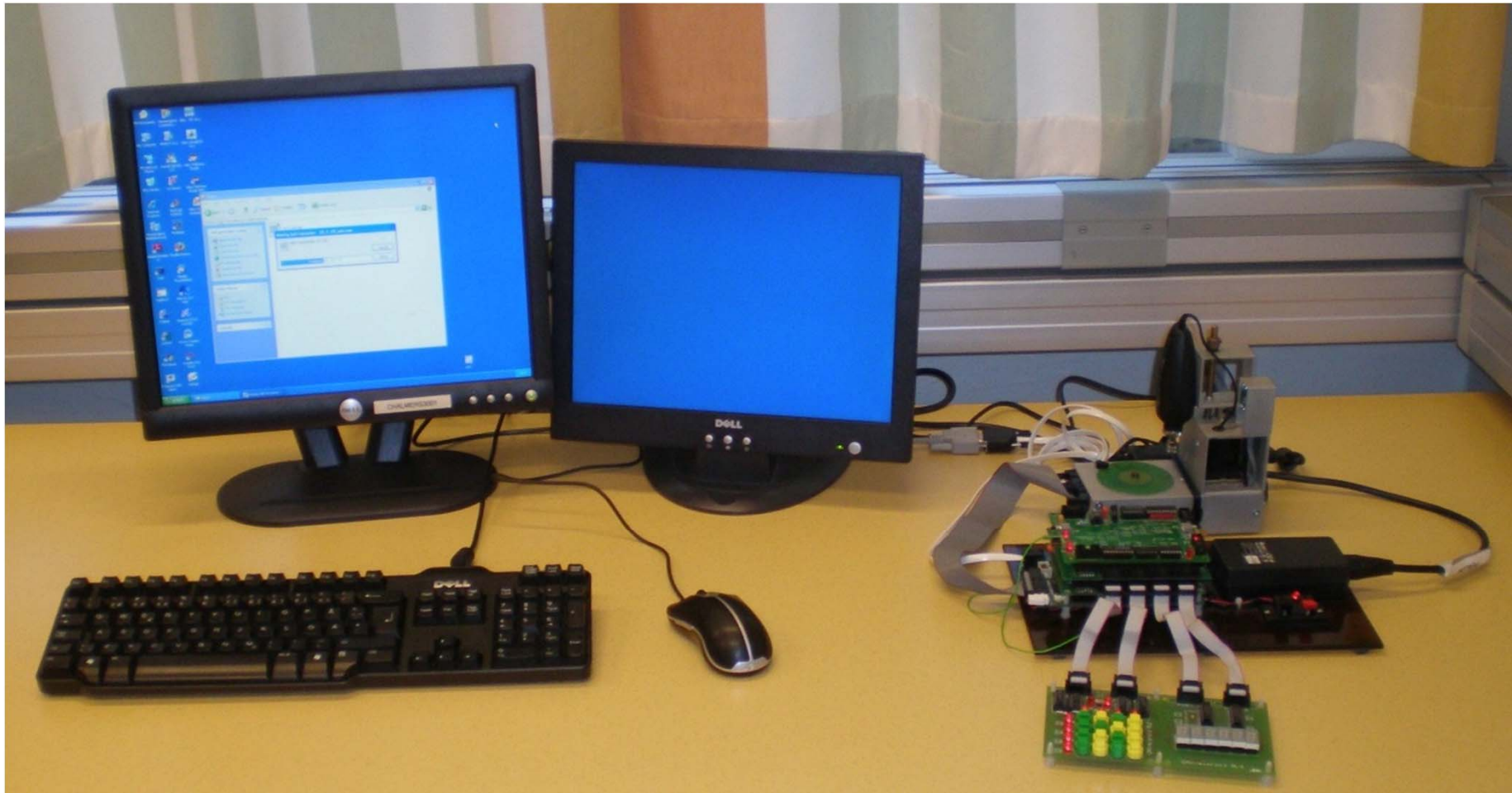
Laborationsöversikt

- ❑ Programutveckling i assembler
 - Moment 1: "Övervakning/styrning av bormaskin"
 - Moment 2: "Pseudoparallell exekvering"

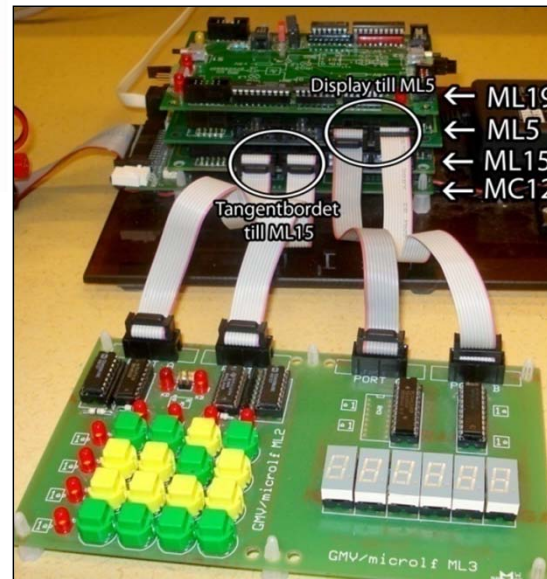
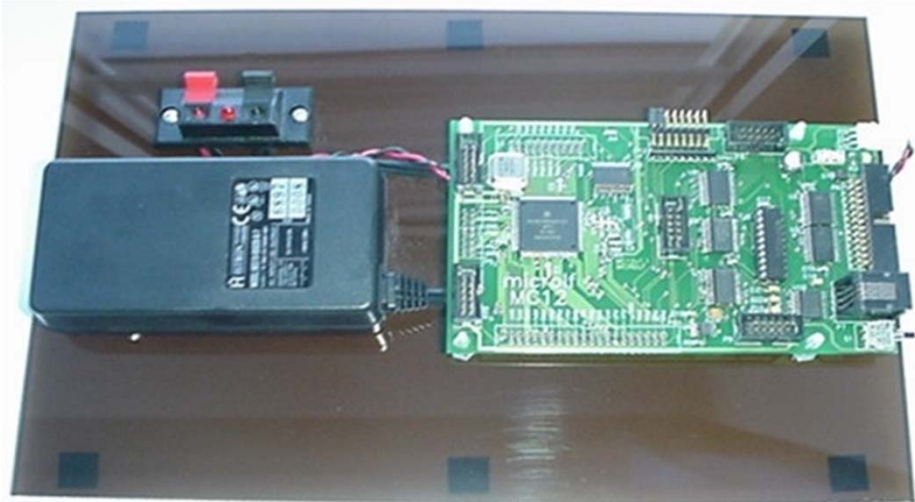
- ❑ Programutveckling i C
 - Moment 3: "Länkad lista med pekare"

- ❑ Maskinnära programmering i C
 - Moment 4: "Övervakning/styrning av bormaskin"

Laborationsplats

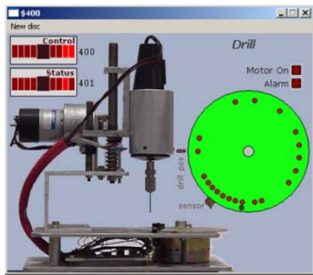


Laborationssystem



Moment 1 och 2:

ETERM för Simulator och laborationssystem



GMV ETERM 6 for MC12

File Edit Debug Windows Help

```

main7.s12
*
#define RUNFAST

        USE        IODEFS.S12

        ORG          Start
        LDS          #BOS

* Calm the drill...
        MOVW        #0,DCtrl
* .. reflect drill status
        MOVW        #0,DCCopy

Loop:
        JSR         KEYB1          ; wait for
        NOP
        JSR         COMMAND ; do instruction
        BRA         Loop

*****
*SUBROUTIN COMMAND
*Beskrivning: Rutinen avgör vilken
*kommandosubrutin som skall anropas och anropar
*denna.
*Anrop:          JSR         COMMAND
*Indata:         Kommandonummer i reg A
*Utdata:         Inga
*Reg-påverkan:  Ingen
*Anrop subr:     SUBO - SUB7
*****

MAX EQU 7
COMMAND PSHA
        PSHX

        CMPA        #MAX
        BHI         COMEX
                
```

MC12 Visual Simulator

Current target setup: drill

Control

Exception handling
 ROM write E/D

Interrupts

Activate X
Service

Status

IO break
 IRQ break
 Running

Program

Address	Instruction	Comment
1000	LDS	#3B00
1003	MOVW	#00,\$0F00
1008	MOVW	#00,\$119C
100D	JSR	\$1037
1010	NOP	
1011	JSR	\$1016
1014	BRA	\$100D
1016	PSHA	
1017	PSHX	
1018	CMPA	#07
101A	BHI	\$1024
101C	ASLA	

Stack

Address	Value
SP (SP)	
3C7A	00
3C7B	00
3C7C	00
3C7D	00
3C7E	00
3C7F	00
3C80	00
3C81	00

Registers

Register	Value	Label
0000	X	
0000	Y	
1000	PC	
1000	SP	
SXHINZVC		
11010000	CCR	

mem

mem	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0010	09	00	00	00	00	00	00	00	00	00	00	00	00	25	00	00
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	08	00	00	F1	00	00	00	00	00

\$9C0

Interface Interrupts

\$400

New disc

Control

400

Status

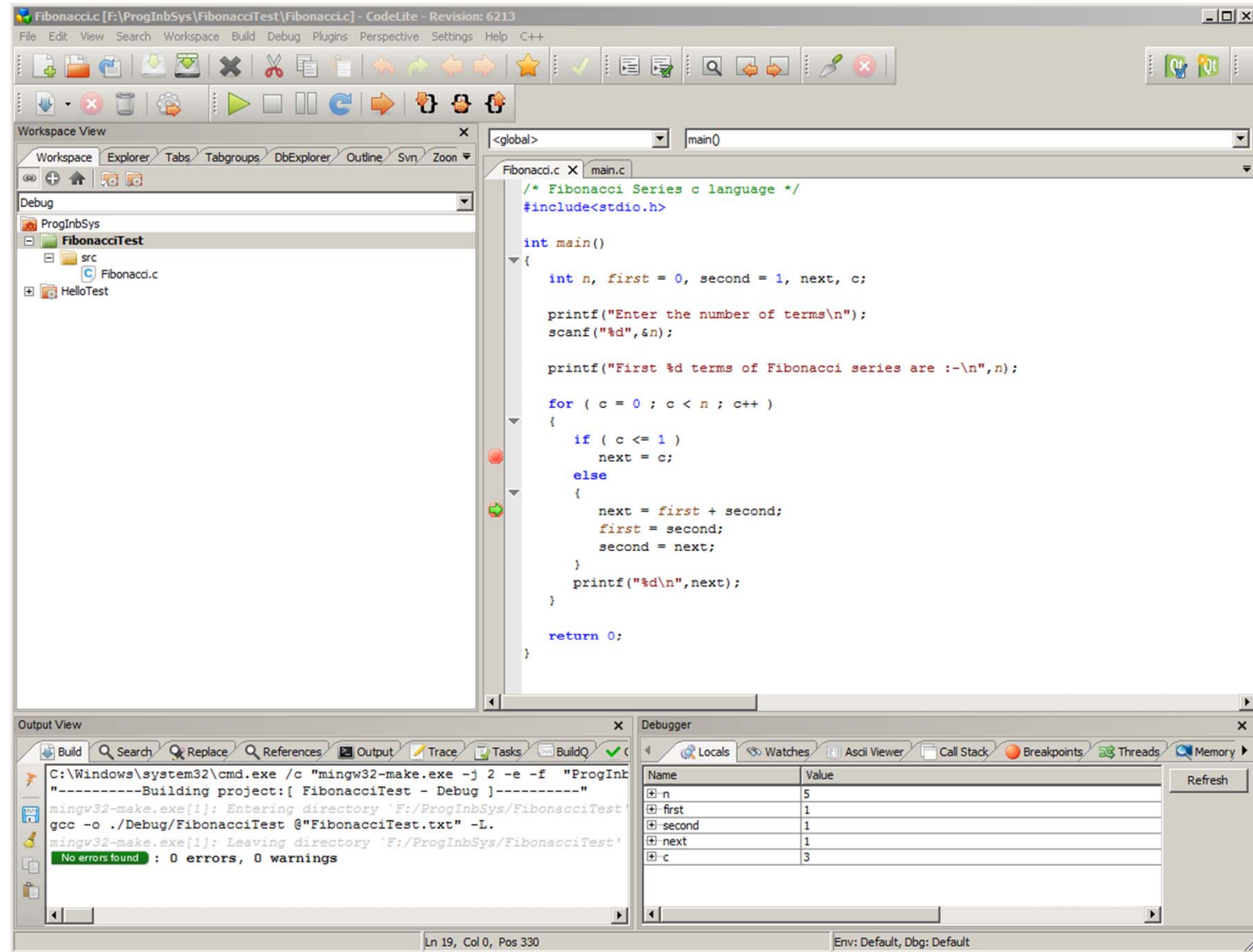
401

Motor On
Alarm

Load Done

Ln 42 Col 14

CodeLite för moment 3 "Länkad lista".



The screenshot displays the CodeLite IDE interface. The main editor shows the source code for a C program that calculates the first 'n' terms of the Fibonacci series. The workspace view on the left shows the project structure, including the source file 'Fibonacci.c'. The output view at the bottom shows the build process, indicating that the program was compiled successfully without errors or warnings. The debugger window at the bottom right shows the current state of variables: 'n' is 5, 'first' is 1, 'second' is 1, 'next' is 1, and 'c' is 3.

```
/* Fibonacci Series c language */
#include<stdio.h>

int main()
{
    int n, first = 0, second = 1, next, c;

    printf("Enter the number of terms\n");
    scanf("%d", &n);

    printf("First %d terms of Fibonacci series are :-\n", n);

    for ( c = 0 ; c < n ; c++ )
    {
        if ( c <= 1 )
            next = c;
        else
        {
            next = first + second;
            first = second;
            second = next;
        }
        printf("%d\n", next);
    }

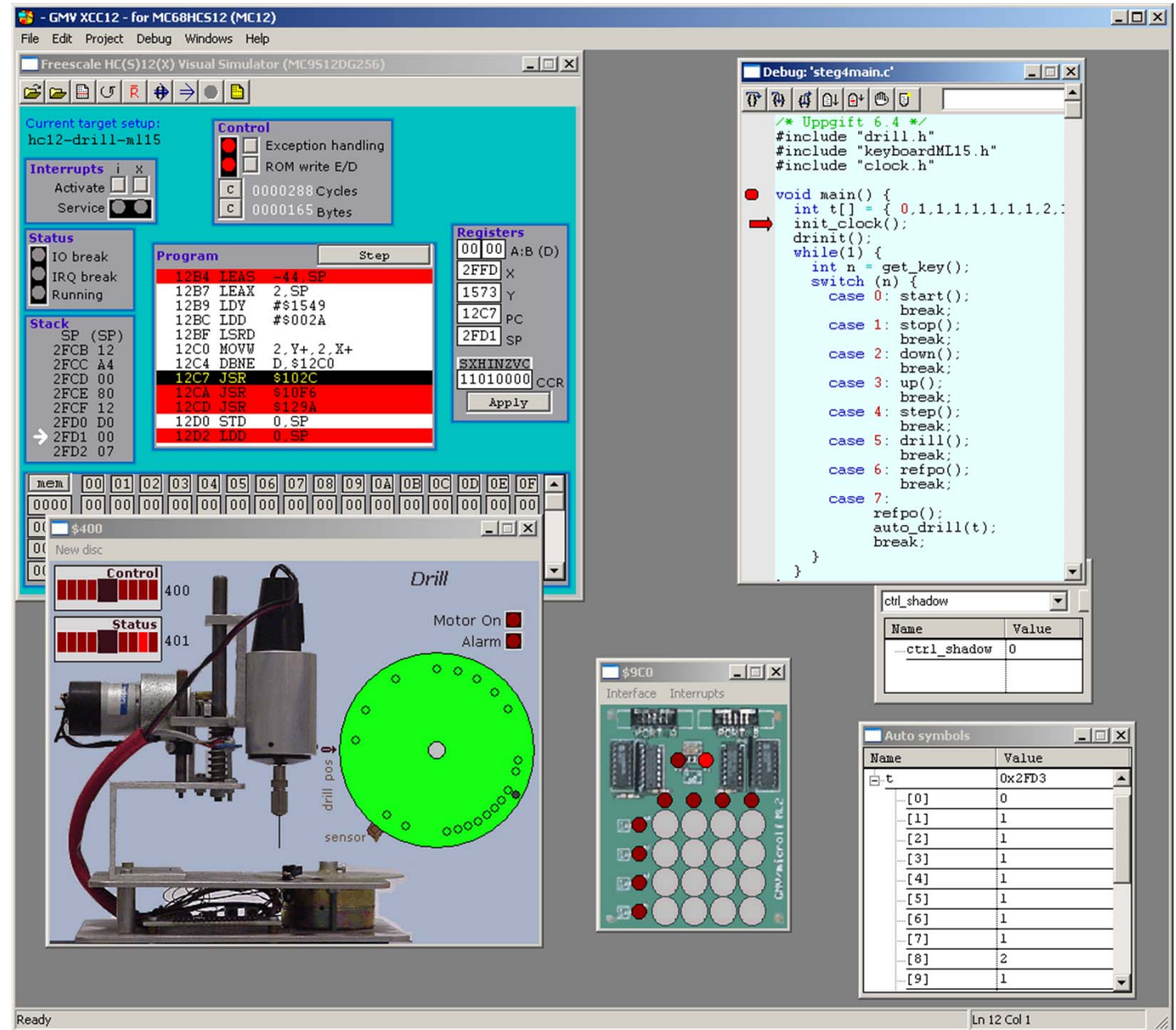
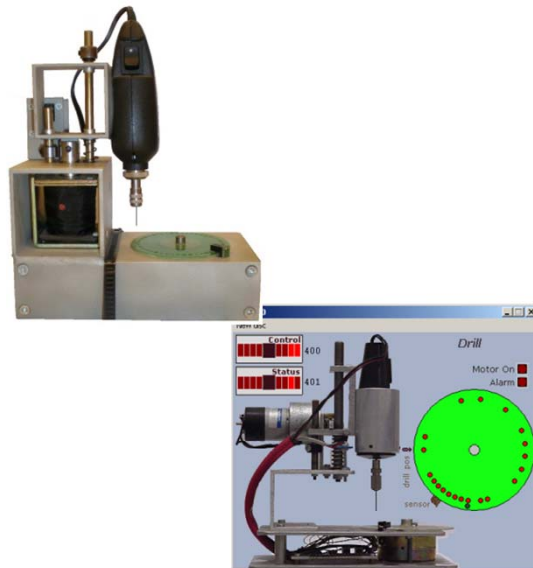
    return 0;
}
```

Build Output:

```
C:\Windows\system32\cmd.exe /c "mingw32-make.exe -j 2 -e -f "ProgInk
-----Building project:[ FibonacciTest - Debug ]-----"
mingw32-make.exe[1]: Entering directory `F:/ProgInbSys/FibonacciTest'
gcc -o ./Debug/FibonacciTest @"FibonacciTest.txt" -L.
mingw32-make.exe[1]: Leaving directory `F:/ProgInbSys/FibonacciTest'
No errors found : 0 errors, 0 warnings
```

Name	Value
n	5
first	1
second	1
next	1
c	3

Moment 4:

XCC12
för Simulator
och laborations-
system

GMV XCC12 - for MC68HC512 (MC12)

Freescalce HC(S)12(X) Visual Simulator (MC9512DG256)

Current target setup: hc12-drill-ml15

Control

- Exception handling
- ROM write E/D
- C 0000288 Cycles
- C 0000165 Bytes

Interruptions

- Activate i X
- Service

Status

- IO break
- IRQ break
- Running

Stack

- SP (SP)
- 2FCB 12
- 2FCC A4
- 2FCD 00
- 2FCE 80
- 2FCF 12
- 2FD0 D0
- 2FD1 00
- 2FD2 07

Program

Address	Instruction	Comment
12B4	LEAS	-44 SP
12B7	LEAX	2, SP
12B9	IDY	#\$1549
12BC	LDD	#\$002A
12BF	LSRD	
12C0	MOVV	2, Y+, 2, X+
12C4	DBNE	D, \$12C0
12C7	JSR	\$102C
12CA	JSR	\$10F6
12CD	JSR	\$129A
12D0	STD	0, SP
12D2	LDD	0, SP

Registers

- 00 00 A:B (D)
- 2FFD X
- 1573 Y
- 12C7 PC
- 2FD1 SP
- SXHINZVC
- 11010000 CCR

Apply

mem 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 \$400

New disc

00

00

Control

- 400

Status

- 401

Drill

Motor On

Alarm

drill pos

sensor

\$9C0 Interface Interrupts

Ctrl Shadow

Name	Value
ctrl_shadow	0

Auto symbols

Name	Value
t	0x2FD3
[0]	0
[1]	1
[2]	1
[3]	1
[4]	1
[5]	1
[6]	1
[7]	1
[8]	2
[9]	1

Ln 12 Col 1

Inför laborationerna

- ❑ Laborationerna måste vara väl förberedda innan laborationstillfället
- ❑ Utveckling och test kan göras med simulatorer
- ❑ Använd kodnings-/simuleringsövningar och hemarbete för förberedelserna
- ❑ ETERM, CodeLite och XCC12 finns på kursens "resurssida", hämta och installera omgående
- ❑ OBS: Laborationerna börjar Måndag i läsvecka 3
ANMÄL ER OMGÅENDE (via kursens hemsida)