

# Programmering av inbyggda system 2014/2015

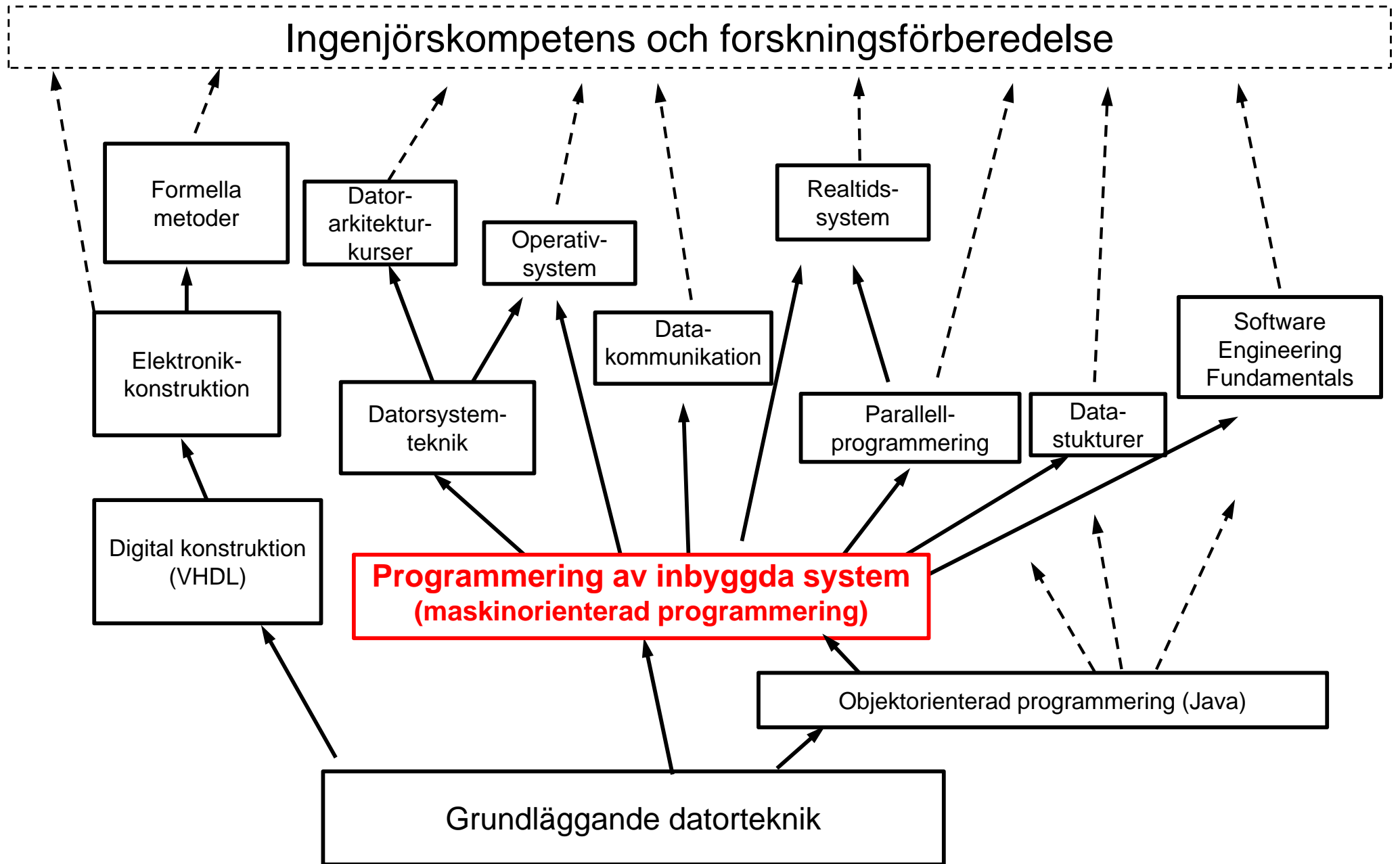
Kursintroduktion  
Roger Johansson

Ur innehållet:

Syften, målsättningar, kurslitteratur och genomförande  
Översikt av laborationer

# Syften och målsättningar

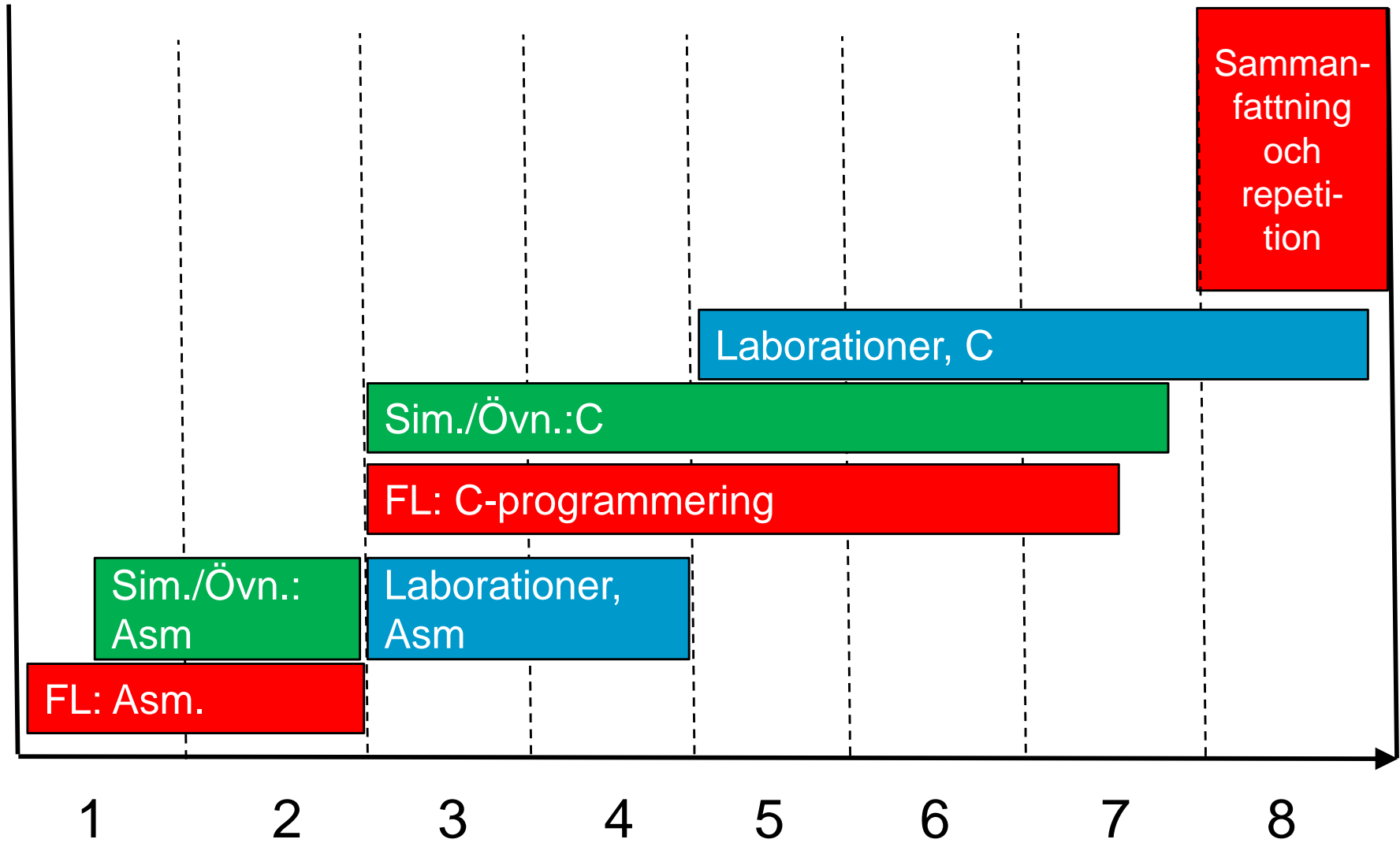
- ❑ Kursens syften är
  - ❑ att vara en introduktion till konstruktion av små inbyggda system och
  - ❑ att ge en förståelse för hur imperativa styrstrukturer översätts till assembler
  - ❑ att ge en förståelse för de svårigheter som uppstår vid programmering av händelsestyrda system med flera indatakällor.
- ❑ 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 ) alt. “The C programming language”
- “Arbetsbok för MC12” ( DC )
- Laborations-PM och inlämningsuppgift,  
finns att plocka upp på DC
- Övrigt material är på elektronisk form och du kan hämta det via kursens hemsida.

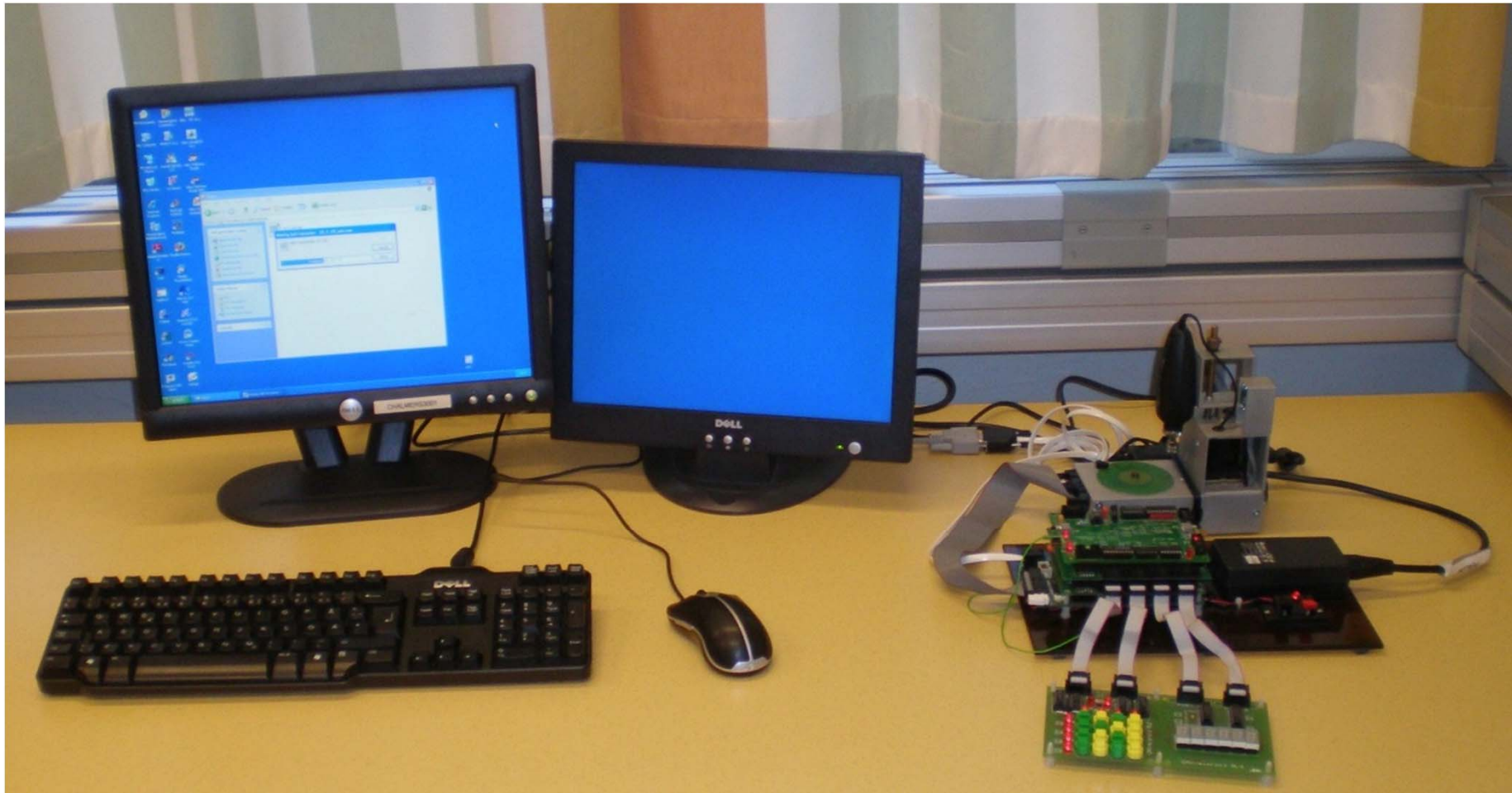
# Genomförande



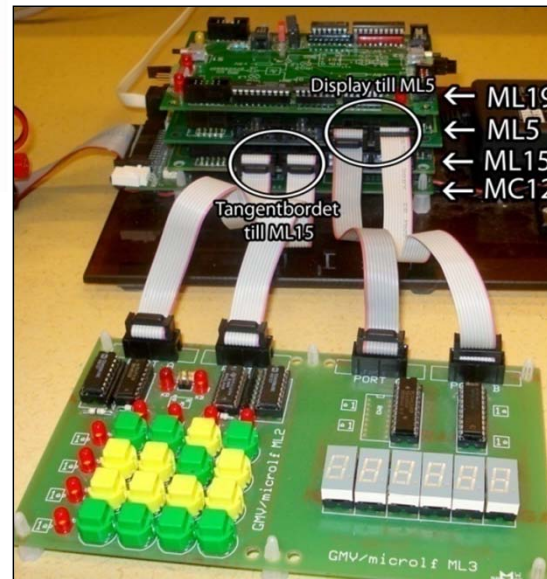
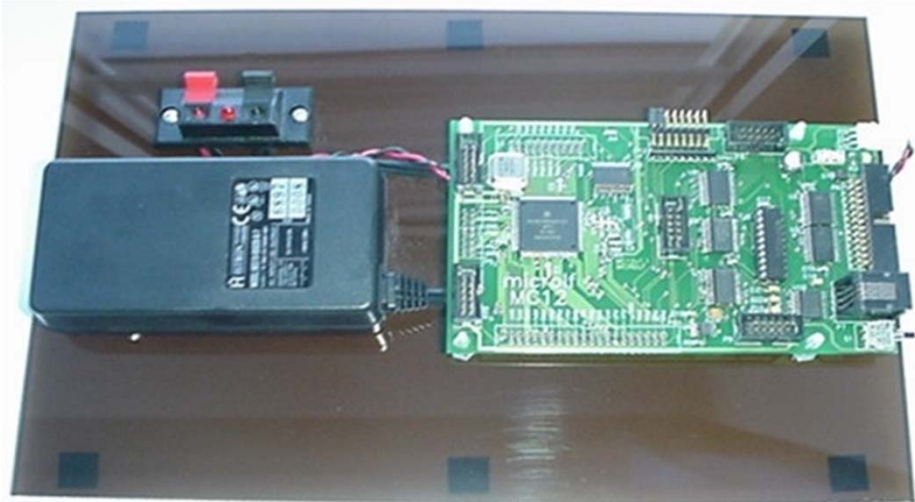
# Laborationsöversikt

- ❑ Moment 1: Inledande programmering i assembler  
"Introduktion till laborationssystemet"
- ❑ Moment 2: Programutveckling i assembler  
"Övervakning/styrning av bormaskin"
- ❑ Moment 3: Programutveckling i assembler  
"Pseudoparallell exekvering"
- ❑ Moment 4: Programutveckling i C  
"Goldbach hypotes" och "Prioritetskö"
- ❑ Moment 5: Maskinnära programmering i C  
"Övervakning/styrning av bormaskin"

# Laborationsplats



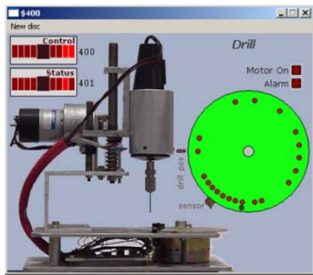
# Laborationssystem





Moment 1, 2 och 3:

## 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...
    MOV     #0,DCtrl
* .. reflect drill status
    MOV     #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** i X

Activate    
Service

**Status**

IO break  
 IRQ break  
 Running

**Program**

Address	Instruction	Step
1000	LDS #3B00	
1003	MOVB #00,\$0F00	
1008	MOVB #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	A:B (D)
0000	Y	
1000	PC	
3C80	SP	
SXHINZVC		
11010000	CCR	

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 25 00 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 00 F1 00 00 00 00

\$400

New disc

**Control**

400

**Status**

401

Motor On   
Alarm

Drill

drill pos

sensor

\$9C0

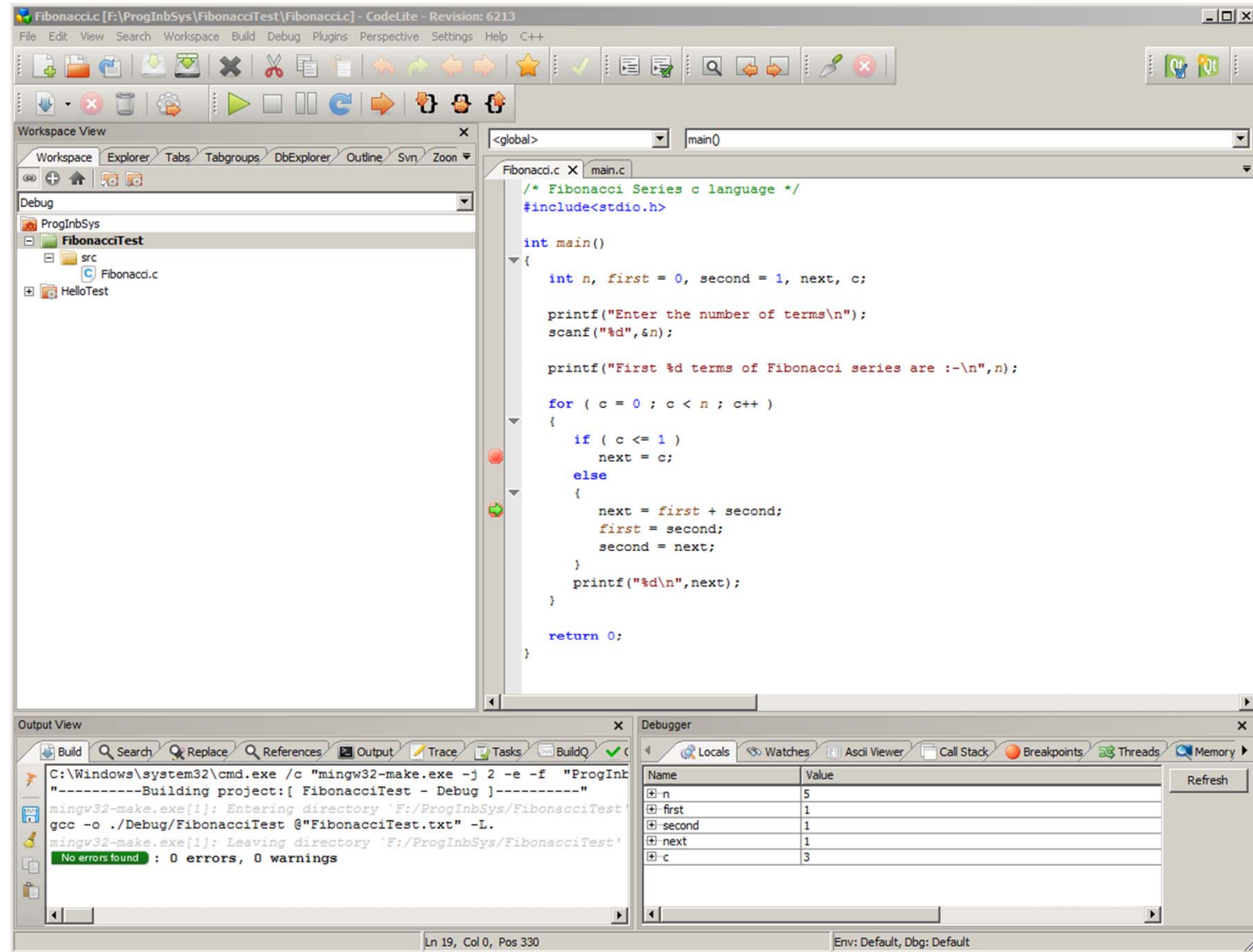
Interface Interrupts

PCNT 0 PCNT 1

CMVMicro1 v1.2

Ln 42 Col 14

CodeLite för  
moment 4  
"Goldbach  
hypotes"  
och  
"Prioritetskö".



```
/* 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;
}
```

Output View

```
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
```

Debugger

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

Ln 19, Col 0, Pos 330  
Env: Default, Dbg: Default



# 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 i läsvecka 3 (efter påsk)  
**ANMÄL ER SENAST ONSDAG LV2  
(via kursens hemsida i PingPong)**