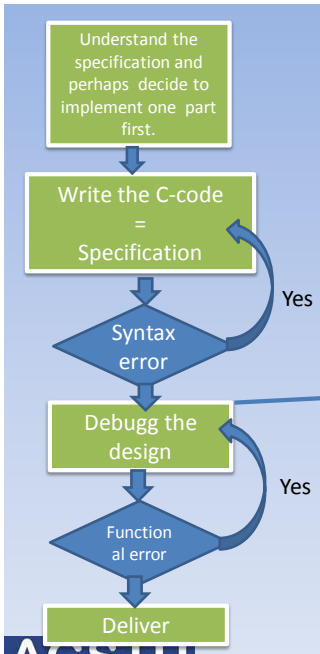
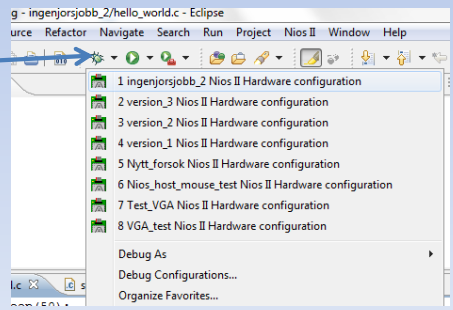
Low-level C-programming and microprocessor architecture Theory 10

- Software architecture
- Work as simple as possible
- Software architecture
- Application architecture
- Examples
- Flow description
- Multitasking introduction

AGSTU Arbete Genom STUdier Utbildning



- WASAP - Work As Simple As Possible



AGSTU Arbete Genom STUdier Utbildning

Software architecture

```
#include  ..
#define  …
// Functions

int main()
{

// Initialization of variables and output to
hardware.


  while (1)
        {

// Here is the program


        }

}
```

AGSTU Arbete Genom STUdier Utbildning

Application architecture

```
#include <task_A_C.c>
main()
{
        main_init();
        while(1)  // loop
for ever
        {
                task_A();
                task_B();
                …
                task_C();
        }
}
```

```
task_A()
{
        task_A_init();
        task_AA();
        task_AB();
                …
        task_AC();
}
Task_B()
….
```

AGSTU Arbete Genom STUdier Utbildning

Periodic start

```
#include <altera_avalon_timer_regs.h> // device
driver for My_Timer

void delay_half_second(void)
{TIMER_RESET;
 TIMER_START;
while (TIMER_READ < 25000000){};
// wait half a second, 50Mhz system clock
}
```

AGSTU Arbete Genom STUdier Utbildning

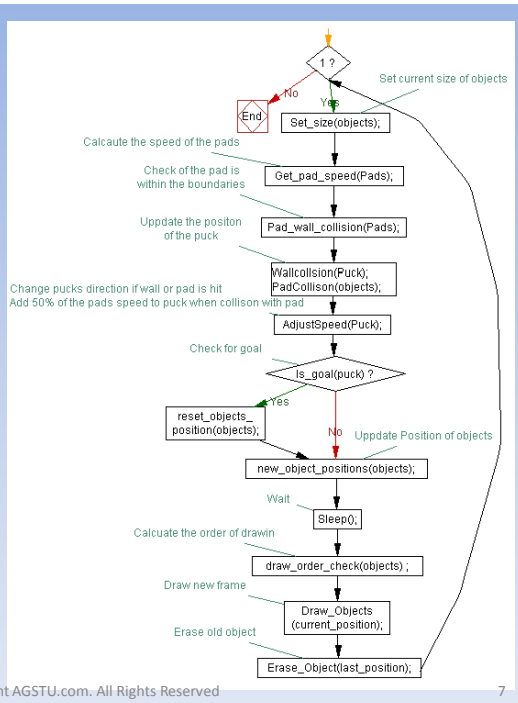
Example State coding

```
#include <…….c>
main()
{
        main_init();
while(1) // loop for ever
        {
        key= key_pressed();
        led(key);
                …
        waite(time);
}
}
```

```
key_pressed()
{
……
  switch (state) {
  case INIT_key_state:
          ……
          break;
  case wait_key_down_state:
          if key….
          state =
       wait_key_up_state;
          break;
  case wait_key_up_state:
          …..
```

AGSTU Arbete Genom STUdier Utbildning

Flow Description

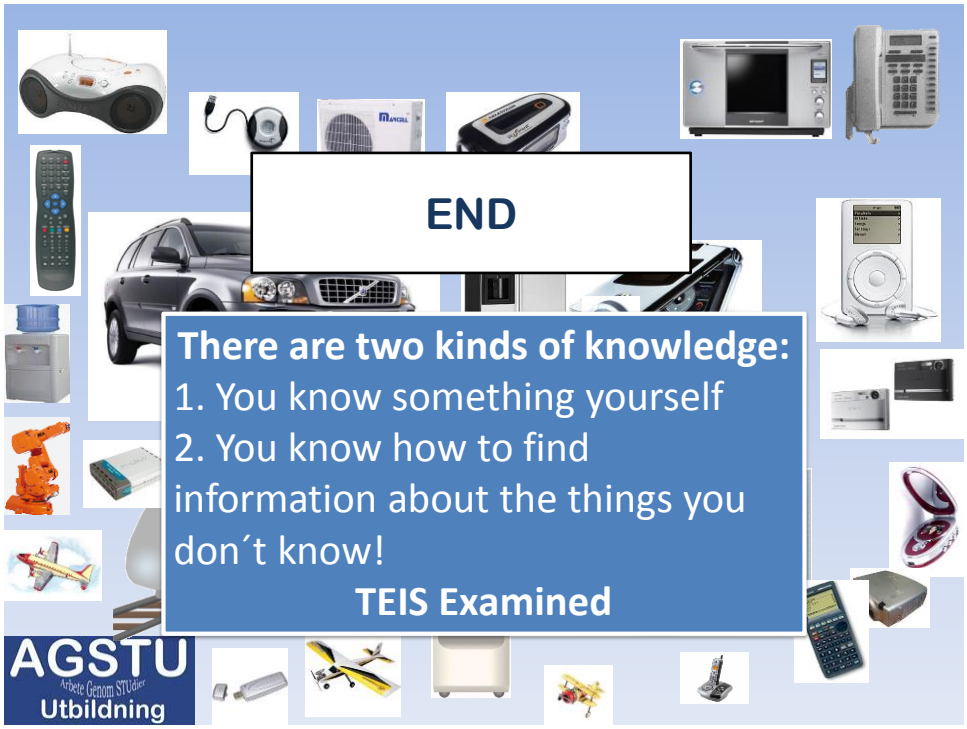


Multitasking introduction

- Executes on the same processor
- Own stack pointer, register data etc.
- More in next course

```
#include …
Task_a()
{
       task_a_init();
while(1) // loop for ever
       { Wait(10ms);
              …..
       }
}
```

```
#include …
Task_b()
{
       task_b_init();
while(1) // loop for ever
       {Wait(20ms);
              …..
       }
}
```

All rights reserved and Disclaim

- **All rights reserved.** No part of this document (PPT, Doc, film etc.) may be reproduced, in any form or by any means, without permission in writing from the publisher. Unless otherwise specified, all information (including software, designs and files) provided are copyrighted by AGSTU AB (www.agstu.se)
- AGSTU AB, Dragverksgatan 138, SE-724 74 Vasteras, Sweden
- **Disclaim**
  All the information (including hardware, software, designs, text and files) are provided "as is" and without any warranties expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose. In no event should the author be liable for any damages whatsoever (including without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use or inability to use information (including text, software, designs and files) provided in this document.

AGSTU
Arbete Genom STUdier
Utbildning