

Robot Design Introduction to NQC - p. 5/17

# LUND UNIVERSITY

# **Program structure 2**

// body

subroutines (no args, no nesting, max 8) sub mysubroutine() {

may be global and local

variables: always 16 bit integers



AI@CS

Department of omputing Science

limited pool: 32 global, 16 local (declare as locally as possible!) arrays declare: int my\_array[3]; use: my\_array[0] = 12; my\_array[2] = my\_array[1]; (note limitations!)

need to be declared (int x; or int y = 2;)

### Statements;

Lund

UNIVERSITY

Al@CS

Department of

- variable declaration;
- assignment;
- control structures
  - if (condition) consequence
  - if (condition) consequence else alternative
  - compound statement ( { . . . }
  - while (condition) body
  - do body while (condition)
  - for (stmt1; condition; stmt2) body
  - repeat (expression) body
  - switch (expression) body
  - **goto label**:



#### Statements 2



access control (prioritisation):

acquire (resources) body acquire (resources) body catch handler

event monitoring:

monitor (events) body monitor (events) body handler list

catch ( catch events ) handler catch handler

Al@CS Department of Computing Science

# Events must be numbered (EVENT\_MASK () macro)

duction to NOC - p. 9/17



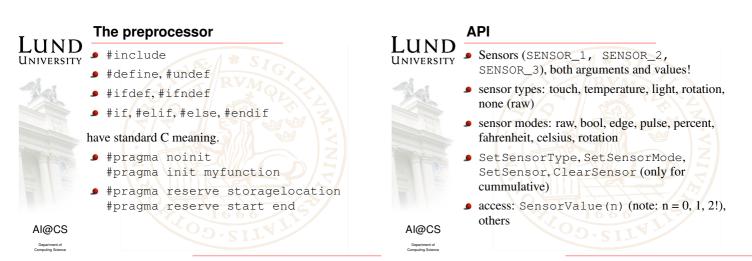
## **Statements 3, Expressions**

- task activation (start, stop)
- break, continue within loops
- expression (only x++; or y- -; make sense)

- values and their combinations by using operators, yield a value
- conditions, yield a logical value (true, false, usual C convention)



Robot Design Introduction to NOC - p. 10/17

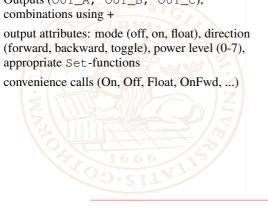


Robot Design Introduction to NQC - p. 11/17

#### API 2

- Lund UNIVERSITY
- Outputs (OUT\_A, OUT\_B, OUT\_C), combinations using +
- output attributes: mode (off, on, float), direction (forward, backward, toggle), power level (0-7), appropriate Set-functions

AI@CS Department of



#### Sound, Display, Communication Lund PlaySound (constant), SOUND\_CLICK, UNIVERSITY SOUND DOUBLE BEEP, SOUND DOWN, SOUND\_UP, SOUND\_LOW\_BEEP, SOUND\_FAST\_UP. PlayTone(int freq, const duration) mute, unmute, clear SelectDisplay (mode). SetUserDisplay (var, const) Message () reads the buffer, ClearMessage() clears it SendMessage(int), SetTxPower(const) Al@CS serial communication is possible Department of omputing Science

Robot Darian Introduction to NOC - p. 14/17

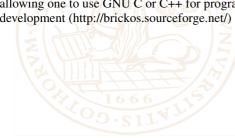
**Timers, Counters Other Stuff** Lund Lund Timers: 4 with 100ms or 10ms resolution; SetPriority (prio) - sets task's priority to ٩ UNIVERSITY UNIVERSITY prio. Useful for access control ClearTimer(0), Timer(2), SetTimer(3, x), FastTimer(2) (acquire-statements) Counters — overlap with memory locations 0-2event monitoring - up to 16 freely configurable (use #pragma reserve!) events logging data: CreateDatalog(size), ClearCounter(1), IncCounter(1), DecCounter(1), Counter(1) AddToDatalog(val) Mait (hundreths) StopAllTasks() Random(const) Al@CS Al@CS sleeping, program switching, battery access, firmware version, clock access Department of Computing Science Department of Computing Science tion Introduction to NOC - p. 15/17 Robot Davian Introduction to NOC - p. 16/17



#### Other systems than NQC

- LeJOS (www.lejos.org), works also for NXT (new generation brick): gives you Java Virtual Machine (but without GC)
  - BrickOS replacement operating system, allowing one to use GNU C or C++ for program development (http://brickos.sourceforge.net/)

Al@CS Department of Computing Science



Robot Design Introduction to NQC - p. 17/17