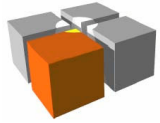


**UMIK**

Usługi mobilne i kontekstowe

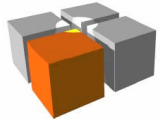
Wykład - 13

Aleksander Pruszkowski



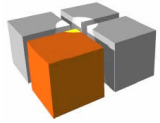
### Lonworks - podstawy działania

- Dziedziny zastosowań
  - HVAC (Heating, ventilation and air-conditioning) i automatyka domowa
  - Sterowanie przemysłowymi procesami produkcyjnymi
  - Systemy zarządzania i monitoringu sieci energetycznych
  - Systemy medyczne
  - Systemy zabezpieczania i nadzoru nad mieniem
  - ...



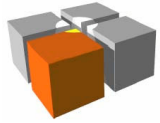
### Lonworks - podstawy działania

- Lista zarejestrowanych producentów: 230 pozycji, m.in.:
  - ABB
  - Alstom Transport SA
  - Danfoss
  - Grundfos Management A/S
  - Hitachi Ltd.
  - Honeywell International, Inc. (12)
  - Hyundai Autonet Co., Ltd.
  - IBM
  - Legrand
  - LG Electronics
  - Mitsubishi Electric Corp.
  - Motorola, Inc.
  - Panasonic Electric Works Co., Ltd.
  - Philips Lighting B.V.
  - Quantum Automation Pte Ltd.
  - Rockwell Automation
  - Samsung Electronics Co., Ltd.
  - Schneider Electric
  - Siemens BT
  - Toshiba Corporation
  - WAGO
  - Yaskawa Electric America
  - ...



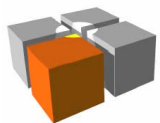
### Lonworks - podstawy działania

- Protokół sieciowy w Lonworks
  - Opisany jako „Control Network Protocol” (CNP)
    - Opracowany przez Echelon Corporation
    - Implementacja CNP została wytworzona przez Echelon i jest nazywana LonTalk
  - Protokół ten jest niezależny od medium transmisyjnego
    - Zapewnia niezawodne i efektywne dostarczanie małych wiadomości przy minimalnym zużyciu pasma transmisyjnego
    - Wykrywa zduplikowane wiadomości
    - Może działać w prostych i tanich w produkcji urządzeniach
      - np.: kompletna implementacja protokołu to kod o wielkości mniejszej niż 10KB korzystająca z mniej niż 1KB danych
    - Jest odpowiednikiem protokołu TCP/IP



### Lonworks - podstawy działania

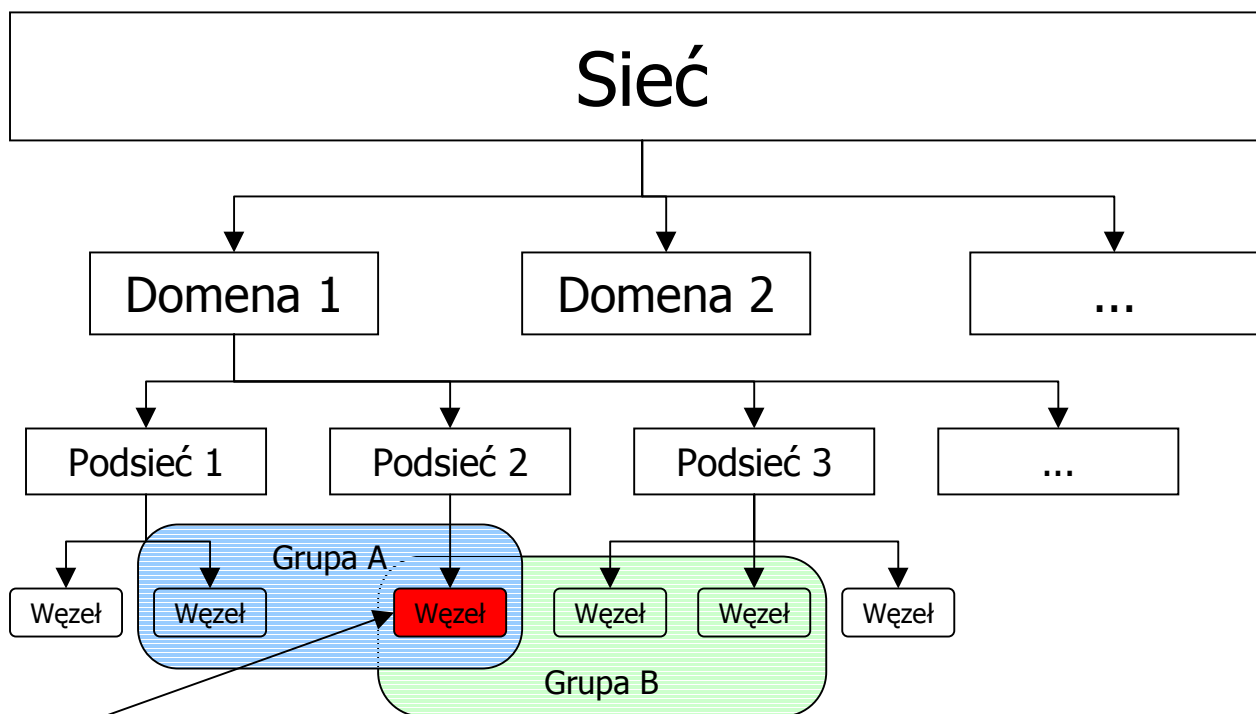
- Warstwa sieciowa Lonworks
  - Jako połączenia fizyczne może wykorzystywać
    - Free Twisted pair wire (TP/FT-10)
      - 78,13Kbps, topologie: magistrali, pętli, gwiazdy oraz ich mix (free topology), zeźnie od źródła zasilania: 64/128 urządzeń w jednym segmencie o długości do 500m
      - odmiana: Transformer-Isolated Twisted Pair (TP/XF-1250) - transfer: 1,25Mbps
    - Power line
      - wersje: A - 3,6Kbps (podnośne: 75KHz-86KHz)/C - 5,4Kbps (podnośne: 115KHz-132KHz)
    - Radio frequency, Fiber optics, Infrared, LonTalk over IP (IP-852)



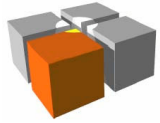
## Usługi mobilne i kontekstowe - Lonworks

### Lonworks - podstawy działania

- Hierarchia urządzeń w sieci

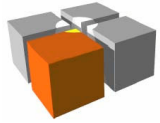


Pojedynczy węzeł może należeć do wielu grup



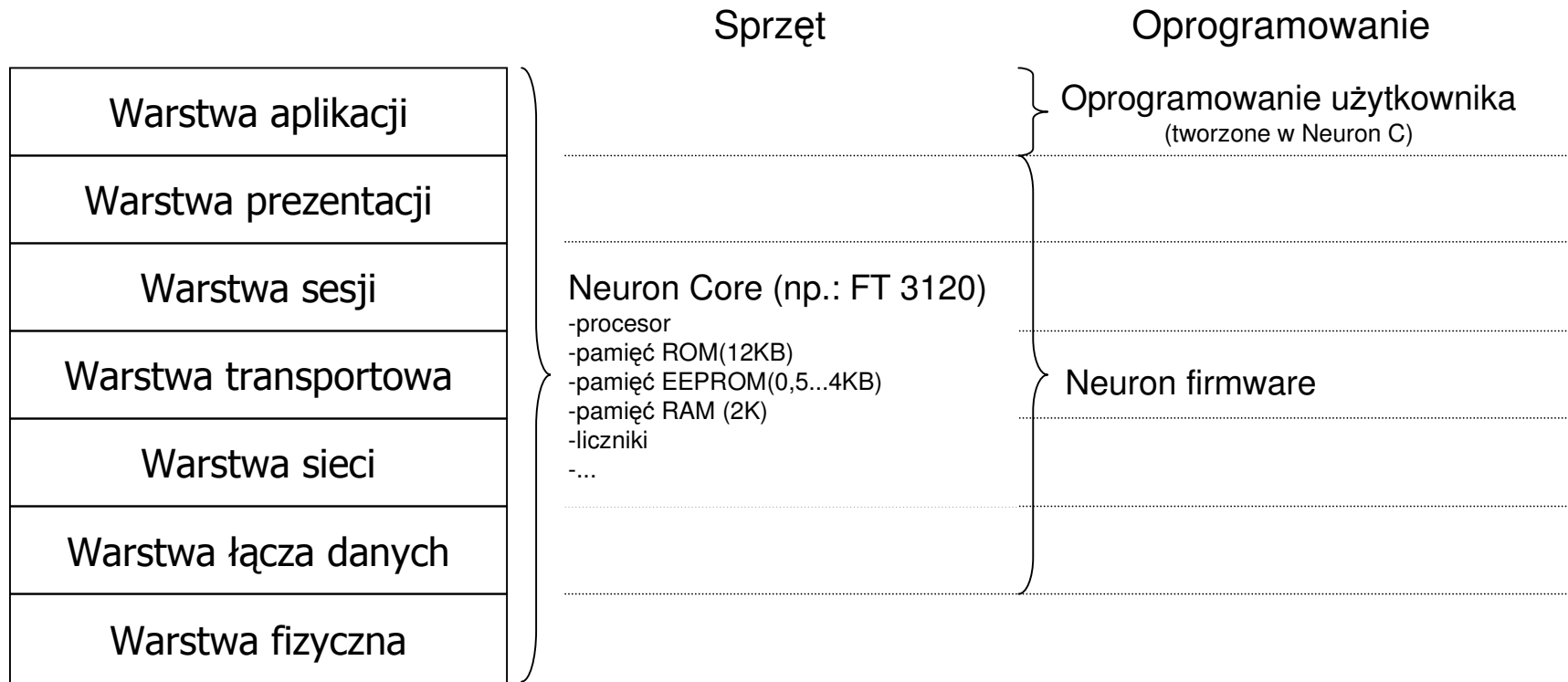
### Lonworks - podstawy działania

- Ograniczenia sieci
  - Do 248 domen w jednej sieci
  - Do 32385 urządzeń w pojedynczej domenie
  - Do 255 podsieci/256 grup w jednej domenie
  - Do 127 urządzeń w jednej podsieci
  - Do 64 urządzeń w jednej grupie (wykorzystujących potwierdzenie)
  - Do 32385 urządzeń w jednej grupie (nie wykorzystujących potwierdzeń)
  - Do 4096 „bindowalnych” zmiennych sieciowych w pojedynczym urządzeniu
    - Faktycznie większość urządzeń może wspierać dużo mniej zmiennych sieciowych (np.: 127)
  - Do 8192 „bindowalnych” aliasów zmiennych sieciowych w pojedynczym urządzeniu

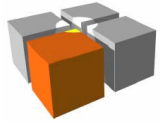


## Lonworks - podstawy działania

### ■ Budowa urządzeń







## Lonworks - podstawy działania

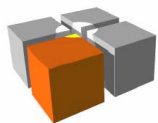
- Neuron C - język programowania urządzeń
  - Baza - ANSI C
  - Zwiera rozszerzenia
    - Własne typy danych, np.: timery

```
mtimer led_on_timer;
```
    - Mechanizmy komunikacji sieciowej, np.: zmienne sieciowe

```
network input SNVT_temp temperature;
```
    - Mechanizmy obsługi peryferii (I/O), np.: podłączenie kontrolnej diody led

```
IO_0 output bit io_display_LED;
```
    - Programowa obsługa zdarzeń, np.: kaskadowa reakcja na zdarzenia

```
...  
when (io_changes){           // Zdarzenie zmiany stanu peryferii (io_changes)  
    // zadanie do wykonania jako reakcja na to zdarzenie  
}  
when (timer_expires){       // Zdarzenie upływu czasu w liczniku (timer_expires)  
    // zadanie do wykonania jako reakcja na to zdarzenie  
}
```



### Lonworks - podstawy działania

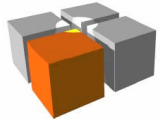
#### ■ Komunikacja sieciowa

##### ■ Format wiadomości warstwy sieci

msb				lsb
Wersja		Format pakietu		Długość
Adres (3-9 bajtów)				
Domena (0/1/3/6 bajtów)				
Treść pakietu				

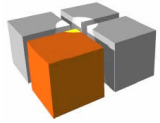
##### ■ Format pakietu wskazuje z jaką usługą powiązać jego treści

- transport (wiadomość do potwierdzenia, potwierdzenie, przypomnienie, przypomnienie-odpowiedź, powtórka)
- sesja (żądanie, odpowiedź, przypomnienie, przypomnienie-odpowiedź)
- uwierzytelnianie (prośba o uwierzytelnienie, odpowiedź uwierzytelniająca)
- prezentacja (wiadomość aplikacji standardowej/użytkowej, ramki diagnostyczne/zarządzania siecią, obca ramka wiadomości, zmienna sieciowa)
  - aplikacje standardowe: logowanie, ISI (auto konfiguracja), Lon FTP, ...



## Lonworks - podstawy działania

- Formaty adresów (tryby adresowania)
  - Format 0
    - Wiadomość dociera do wszystkich urządzeń w zaadresowanej podsieci, rozmiar adresu:  $2B(\text{podsieć nadawcy i adres nadawcy}) + 1B(\text{ID podsieci})$
  - Format 1
    - Wiadomość dociera do wszystkich urządzeń w zaadresowanej grupie, rozmiar adresu:  $2B(\text{podsieć nadawcy i adres nadawcy}) + 1B(\text{ID grupy})$
  - Format 2a
    - Wiadomość dociera do określonej podsieci i określonego w niej węzła, rozmiar adresu:  $2B(\text{podsieć nadawcy i adres nadawcy}) + 2B(\text{ID podsieci i ID węzła})$
  - Format 2b
    - Wiadomość dociera do określonej podsieci i określonego w niej węzła, rozmiar adresu:  $2B(\text{podsieć nadawcy i adres nadawcy}) + 3B(\text{ID podsieci i ID węzła} + \text{ID grupy odbiorcy})$
  - Format 3
    - Wiadomość dociera do określonego węzła (Neuron chip), rozmiar adresu:  $2B(\text{podsieć nadawcy i adres nadawcy}) + 1B(\text{ID podsieci}) + 6B(\text{Neuron ID})$ 
      - używany gdy węzły nie zostały jeszcze skonfigurowane

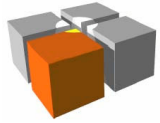


## Usługi mobilne i kontekstowe - Lonworks

---

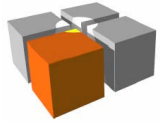
### Lonworks - podstawy działania

- Sposoby komunikacji
  - Zmienne sieciowe
    - Zalecany sposób wymiany danych
  - Wiadomości aplikacji
    - Częstki aplikacji pochodzącej od jednego dostawcy mogą się ze sobą komunikować za pomocą własnych typów wiadomości
    - Użyteczny w wielkich aplikacjach
  - Właściwości konfiguracyjne
    - Używane przez narzędzia do konfigurowania zdalnego aplikacji w urządzeniu
  - „Obce ramki”
    - Używane w komunikacji z systemami zewnętrznymi
  - Powyższe metody komunikacji nie wykluczają się wzajemnie!



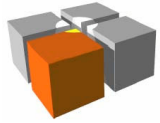
### Lonworks - podstawy działania

- Zmienne sieciowe (NV)
  - Zmienne są zasobem umożliwiającym wymianę treści między współpracującymi w węzłach aplikacjami
    - Każde urządzenie może używać jedną lub wiele zmiennych sieciowych
  - Zmienne sieciowe mają:
    - Kierunek (wejściowe - np.: związane z elementami wykonawczymi, wyjściowe - np.: związane z sensorami)
    - Typ
      - określa kodowanie i jednostki w jakich zapisano treść zmiennej np.: stopnie C/F
      - określa możliwości współpracy urządzeń - np.: wyklucza łączenie termometru generującego odczyty w stopniach C z urządzeniem grzewczym posługującym się stopniami F
      - mogą reprezentować proste wartości lub struktury
    - Długość: 1B...31B



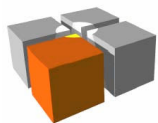
### Lonworks - podstawy działania

- Zmienne sieciowe (NV)
  - Fizyczne umieszczenie zmiennych sieciowych
    - Czy to ważny aspekt?
    - Repliki zmiennych domyślnie umieszczane są w RAM węzła
      - stan po włączeniu nie jest podlega standaryzacji
      - aplikacja po przywróceniu zasilania musi je uaktualnić - mogą powstawać wtedy zatory komunikacyjne
    - Repliki zmiennych sieciowych mogą być umieszczone w pamięciach trwałych
      - zanik zasilania nie niszczy ich treści
      - ich częsta zmiana niszczy struktury krzemowe procesora Neuron C (jego pamięci EEPROM/FLASH)



### Lonworks - podstawy działania

- Zmienne sieciowe (NV)
  - Tryb aktualizowania zmiennych sieciowych
    - Procedura aktualizowania zmiennych realizowana jest przez Neuron firmware
      - domyślnie system aktualizuje najszybciej jak to jest możliwe, optymalizując liczbę przekazywanych siecią danych
      - wada: gdy zmiany wykonane zostaną zbyt często - system może odtworzyć w innych węzłach tylko ostatnią wartość
      - podejście dobrze sprawdza się gdy przesyłane są wartości absolutne
      - podejście nie sprawdza się jednak przy przesyłaniu wartości relatywnych - zgubienie pojedynczej wartości uszkadza logikę aplikacji
    - Możliwe jest ustalenie trybu synchronicznego pracy zmiennych
      - wszystkie kolejne operacje na zmiennej zostaną odwzorowane w związanych urządzeniach
      - duży narzut komunikacyjny



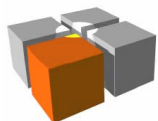
# Usługi mobilne i kontekstowe - Lonworks

## Lonworks - podstawy działania

- Lonmark utrzymuje listę typów standardowych zmiennych sieciowych (Standard Network Variable Types - SNVT)
  - Obecnie zdefiniowano przeszło 200 typów (spis na dostępny na: <http://types.lonmark.org/index.html>):

SNVT_abs_humid	SNVT_chlr_status	SNVT_date_time	SNVT_enthalpy	SNVT_time_val_2
SNVT_address	SNVT_clothes_w_a	SNVT_defr_mode	SNVT_enthalpy_d	SNVT_time_zone
SNVT_alarm	SNVT_clothes_w_c	SNVT_defr_state	SNVT_environment	SNVT_tod_event
SNVT_alarm_2	SNVT_clothes_w_m	SNVT_defr_term	SNVT_evap_state	SNVT_trans_table
SNVT_amp	SNVT_clothes_w_s	SNVT_density	SNVT_ex_control	SNVT_turbidity
SNVT_amp_ac	SNVT_color	SNVT_density_f	SNVT_file_pos	SNVT_turbidity_f
SNVT_amp_ac_mil	SNVT_color_2	SNVT_dev_c_mode	SNVT_file_req	SNVT_valve_mode
SNVT_amp_f	SNVT_config_src	SNVT_dev_fault	SNVT_file_status	SNVT_vol
SNVT_amp_mil	SNVT_count	SNVT_dev_maint	SNVT_fire_indcte	SNVT_vol_f
SNVT_angle	SNVT_count_32	SNVT_dev_status	SNVT_fire_init	SNVT_vol_kilo
SNVT_angle_deg	SNVT_count_f	SNVT_earth_pos	SNVT_fire_test	SNVT_vol_mil
SNVT_angle_f	SNVT_count_inc	SNVT_elapsed_tm	SNVT_flow	SNVT_volt
SNVT_angle_vel	SNVT_count_inc_f	SNVT_elec_kwh	SNVT_flow_dir	SNVT_volt_ac
SNVT_angle_vel_f	SNVT_ctrl_req	SNVT_elec_kwh_l	SNVT_flow_f	SNVT_volt_dbmv
SNVT_area	SNVT_ctrl_resp	SNVT_elec_whr	SNVT_flow_mil	SNVT_volt_f
SNVT_btu_f	SNVT_currency	SNVT_elec_whr_f	SNVT_flow_p	SNVT_volt_kilo
SNVT_btu_kilo	SNVT_date_cal	SNVT_ent_opmode	SNVT_freq_f	SNVT_volt_mil
SNVT_btu_mega	SNVT_date_day	SNVT_ent_state	SNVT_freq_hz	SNVT_Wm2_p
SNVT_char_ascii	SNVT_date_event	SNVT_ent_status	SNVT_freq_kilohz	SNVT_zerospan





## Lonworks - podstawy działania

### ■ Przykładowa SNVT\_switch

#### Overview:

Switch (value, state)

A structure reporting a percentage level or load value and a discrete on/off state. Separate fields report the percentage value and state. This type should be used for both discrete (on/off) and analog control.

The value field is used to control the load's value, i.e. position, speed, or intensity, the state field being used to control whether the load is on or off (enabled or disabled).

When used as the output of a discrete sensor device, the OFF state is represented by a SNVT\_switch network variable with state = FALSE and value = 0. The other discrete states are represented by state = TRUE and value > 0. When used as the output of a two-state sensor device, the ON state is represented by state = TRUE and value = 200 (meaning 100%).

When used as the input of a two-state discrete actuator, a SNVT\_switch network variable with state = TRUE will be interpreted as the ON state if value > 0, and as the OFF state if value = 0. Additionally, a SNVT\_switch input network variable with state = FALSE should be interpreted as the OFF state, whether or not value = 0. A state value of 0xFF indicates the switch value is undefined.

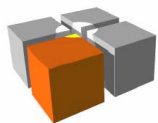
#### Details:

Resource Set: *Standard 00:00:00:00:00:00:00:00*  
 Index: *95*  
 Obsolete: *no*  
 Size: *2*  
 Programmatic Name: *SNVT\_switch*  
 Neuron C Type:

Structure	
value	Value (% of full level) .  unsigned short Minimum: <i>0</i> Maximum: <i>200</i> Scaling (A,B,C): <i>5, -1, 0</i> Scaled value: $5 * 10^{-1} * (Raw+0)$ Resolution: <i>0.5</i>
state	State (state code) . This field can either be -1 (NULL), 0 (OFF), or 1 (ON).  signed short Minimum: <i>-1</i> Maximum: <i>1</i> Invalid: <i>-1</i> Scaling (A,B,C): <i>1, 0, 0</i> Scaled value: $1 * 10^0 * (Raw+0)$ Resolution: <i>1</i>

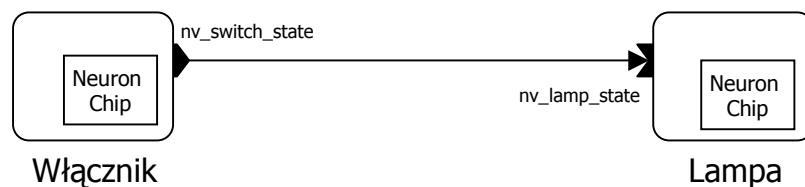
Formats: *SNVT\_switch: text("%f%d", value, state)*

Input Network Variable			Output Network Variable		
value (raw)	state	2-state interpretation	value (raw)	state	2-state interpretation
any	0	off (0; 0)	0	0	off
0	1	off (0; 1)	200 (0xC8)	1	on
> 0	1	on (200; 1)	0 .. 200 (0 .. 0xC8)	-1 (0xFF)	invalid (no action)
any	-1 (0xFF)	invalid (no action)	(any valid value)		

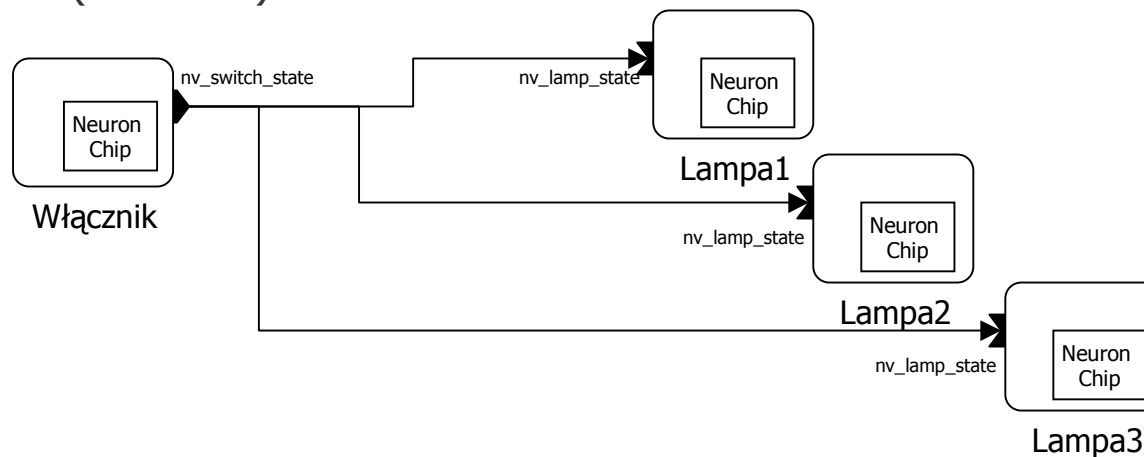


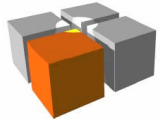
## Lonworks - podstawy działania

- Połączenia zmiennych sieciowych
  - Indywidualne (unicast)



- Grupowe (multicast)





## Lonworks - podstawy działania

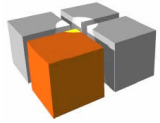
- Deklaracja i użycie zmiennych sieciowych w Neuron C

- Włącznik

```
...
IO_4 input bit io_switch_in;           //określa jak CPU widzi włącznik
network output SNVT_switch nv_switch_state;
when (io_changes(io_switch_in)) {     //czy zdarzenie zmiany stan włącznika
    nv_switch_state.state = (io_switch_in == SWITCH_ON) ? ST_ON : ST_OFF;
}
```

- Lampka

```
...
network input SNVT_switch nv_lamp_state;
when (nv_update_occurs(nv_lamp_state)) { //czy zmodyfikowano zmienną sieciową
    io_out(ioLED, nv_lamp_state.state); //aktualizacja stanu lampy (tu: LED)
}
```

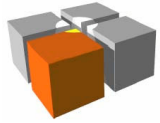


## Usługi mobilne i kontekstowe - Lonworks

---

### Lonworks - podstawy działania

- Wiadomości aplikacji
  - Mogą uzupełniać komunikację gdy urządzenia nie muszą współpracować ze sobą w standardowy sposób
    - np.: grupa urządzeń jednego dostawcy chce komunikować się w zrozumiały tylko sobie sposób
  - Podejście to oszczędza miejsce w pamięci kodu urządzenia - kosztem elegancji i efektywności kodowania
    - Faktycznie zmienne sieciowe bazują na tym mechanizmie do propagacji treści między urządzeniami
  - API
    - wysyłanie/anulowanie/wysyłanie wiadomości: `msg_send()/msg_cancel()/msg_receive()`
    - zdarzenia: `msg_arrives/msg_completes/msg_succeeds/msg_fails`
    - zarządzanie buforami dla wiadomości: `msg_alloc()/msg_alloc_priority()/msg_free()/resp_alloc()/resp_free()`



### Lonworks - podstawy działania

- Wiadomości aplikacji, cd.
  - W treści wiadomości aplikacja ustala:
    - priorytet wiadomości - wiadomość zwykła/priorytetowa
    - znacznik wiadomości (tag) - punkt komunikacyjny aplikacji, odpowiednik identyfikatora wirtualnego połączenia
    - kod wiadomości - pozwalają ustalić czym jest wiadomość
      - wiadomością danej aplikacji/treścią zmiennej sieciowej/wiadomościami systemu zarządzania/...
    - czy wiadomość wymaga uwierzytelnienia
    - tryb realizacji usługi - wiadomość potwierdzana/niepotwierdzona/powtarzana/...
    - adres odbiorcy - pole opcjonalne(!) - omijane wymusza wyznaczenie celu dostarczenia wiadomości na bazie jej znacznika
    - dane - niemal dowolna treść do 228B

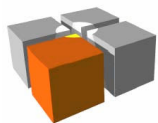


## Lonworks - podstawy działania

- Wiadomości aplikacji, cd.

- Przykład:

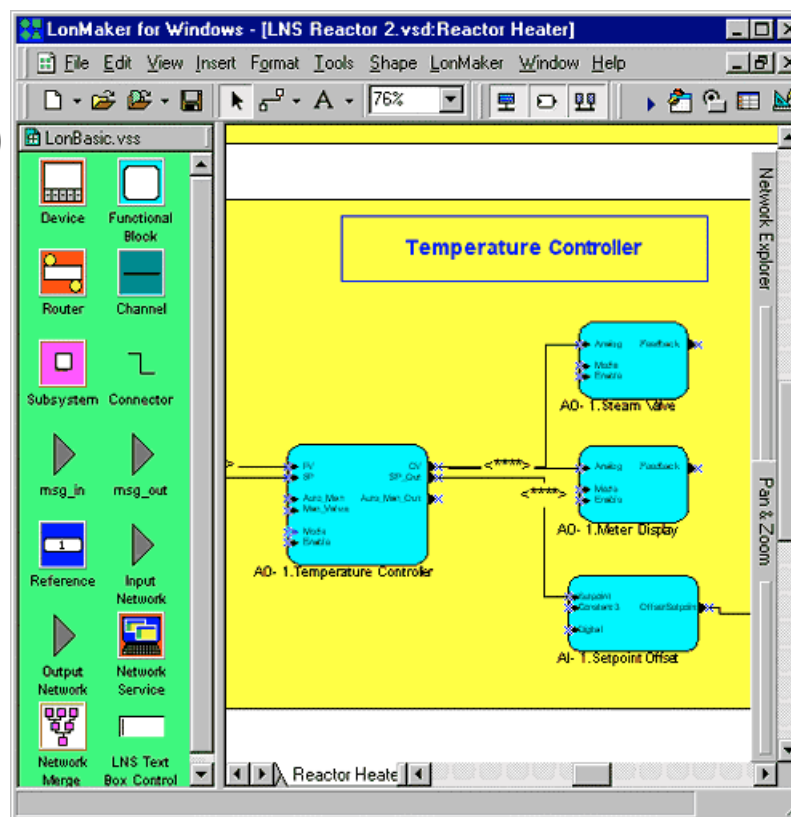
```
msg_tag motor;
#define MOTOR_ON 0
typedef enum { MOTOR_FWD, MOTOR_REV } motor_dir;
struct {
    long motor_speed;
    motor_dir motor_direction;
    int motor_ramp_up_rate;
} motor_on_message;
when(io_changes(...)){
    msg_out.tag = motor;
    msg_out.code = MOTOR_ON;
    motor_on_message.motor_direction = MOTOR_FWD;
    motor_on_message.motor_speed = 500;
    motor_on_message.motor_ramp_up_rate = 100;
    memcpy(msg_out.data, &motor_on_message, sizeof (motor_on_message));
    msg_send();
}
```

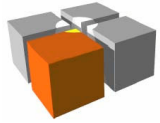


## Usługi mobilne i kontekstowe - Lonworks

### Lonworks - podstawy działania

- Narzędzia
  - Kompilator Neuron C
  - Narzędzie dla integratorów (LonMaker)
    - Koszt ~870/1300\$ (2012.05.31)  
zestaw standardowy/profesjonalny
  - Zestawy startowe (evaluation kit)
    - Koszt od ~400\$  
(zawiera Neuron C, bez LonMaker)





## Lonworks - podstawy działania

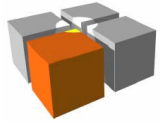
### ■ Narzędzia

#### ■ Biblioteka ISI (Interoperable Self-Installation)

- Podejście dla małych sieci (<200 urządzeń), umożliwia rezygnację z kosztownych narzędzi
- Idea: zarządzanie urządzeniami i ich połączeniami na poziomie aplikacji, przykład:

```
#pragma num_alias_table_entries 6           // włączenie mechanizmów ISI (dla 6 aliasów)
#include <isi.h>                             // dołączenie prototypów ISI
when (reset) {                               // Czyszczenie tablic i uruchomienie modułu ISI
    IsiStartS(isiFlagNone);
}
mtimer repeating isiTimer = 1000ul / ISI_TICKS_PER_SECOND;
when (timer_expires(isiTimer))              // Cykliczne pobudzanie modułu ISI - prace tła
    IsiTickS();
when (msg_arrives) {
    if (IsiApproveMsg())                    // Obsługa wiadomości ISI od innych urządzeń
        (void) IsiProcessMsgS();
}
```





## Usługi mobilne i kontekstowe - Lonworks

---

### Literatura:

- Echelon Corporation, „Neuron(R) C Programmer's Guide - Revision 7”, 2003
- Echelon Corporation, „Introduction to the LonWorks(R) Platform - Revision 2”, 2009
- Echelon Corporation, „NodeBuilder(R) User's Guide”, 2001
- Echelon Corporation, „ISI Programmer's Guide Version 3”, 2007