Università degli Studi di PISA

Anno Accademico 2002/2003

Dipartimento di Informatica,
corso: Sistemi di Gestione di Reti

Progetto di MIB per la gestione di caldaie murali a gas

Gruppo : Andrea Bernardi
Roberto Rossi

Sommario

1. INTRODUZIONE

Le “living room” sono una delle massime espressioni della ricerca, in particolare delle tecnologie informatiche, applicate alla vita dell’uomo nell’ambiente a lui più familiare, la casa.
Quest’ultime infatti sono una delle più recenti innovazioni in campo immobiliare/abitativo e consistono nell’ installazione di sistemi per il controllo in remoto di elettrodomestici, allarmi, porte, finestre, ecc. , all’interno di un’abitazione, arrivando anche al totale controllo di essa.
Le “living room” sono nate in risposta ad un’utenza sempre più esigente in termini di automatizzazione e praticità d’uso degli apparecchi, bisogno di sicurezza e risparmio delle risorse energetiche (vedi sfruttamento dell’energia solare, eolica, etc.).
Quindi lo sviluppo e l’applicazione di suddette tecnologie potranno in futuro (anche prossimo) cambiare radicalmente tutte quelle azioni quotidiane che si svolgono all’interno delle nostre abitazioni, modificando in buona parte l’attuale approccio dell’uomo verso la propria dimora.

L’applicazione delle tecnologie informatiche nell’ uso quotidiano degli elettrodomestici è il naturale contesto in cui si pone il nostro progetto: la gestione tramite MIB SNMP di caldaie murali a gas.
Abbiamo voluto sviluppare un MIB SNMP per un facile controllo ed utilizzo di una caldaia murale a gas concentrando il nostro lavoro su diversi aspetti, da quello più semplice, come lo stato della caldaia (se è accesa o spenta), fino ad arrivare ai controlli dei valori riguardanti pressione,

temperatura e livello dei liquidi nell'impianto, cercando di abbracciare in maniera più semplice possibile(cioè in pieno stile SNMP) tutte le più comuni funzionalità di una caldaia murale a gas, non tralasciando il rilevamento di tutti i malfunzionamenti, sia critici che non, e di tutti i cambiamenti di stato che si possono incontrare utilizzando un apparecchio del genere.
Tutto questo lo abbiamo fatto traendo spunto da una caldaia murale a gas tuttora realmente installata e funzionante.

2. SVOLGIMENTO

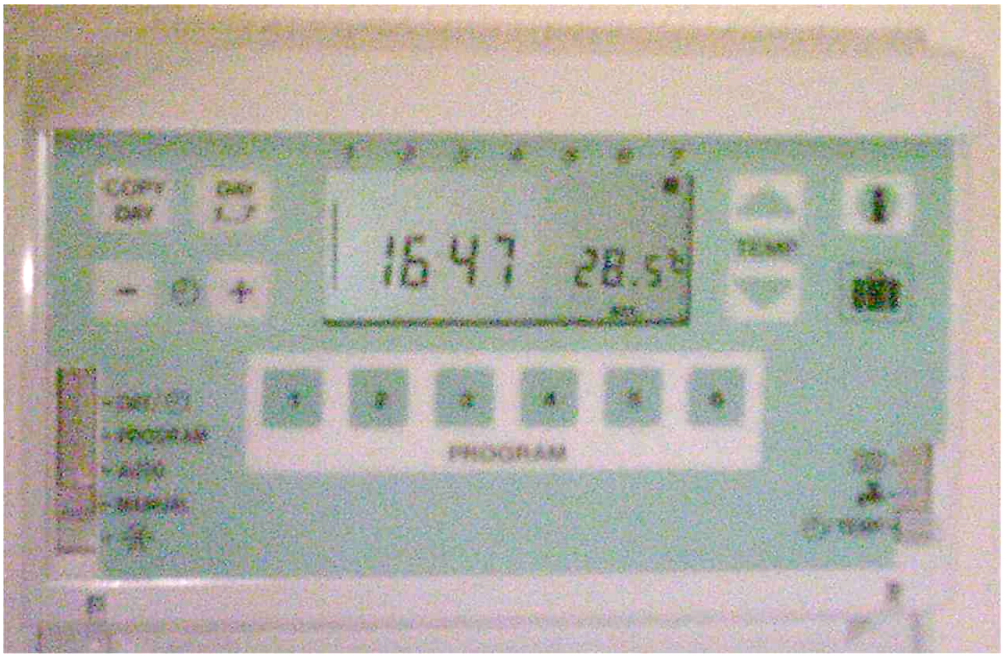
Questo MIB è stato creato sia per l'interrogazione ed il controllo di tutti quei parametri che regolano il funzionamento di una caldaia, sia per rendere effettive le impostazioni desiderate dall'utente (richieste in modo manuale o automatico).
Le variabili usate sono identificative di quelle funzionalità che abbiamo ritenuto fossero necessarie per il buon funzionamento di un boiler.
Potremo quindi verificare lo stato di funzionamento (on /off) sia del circuito di riscaldamento che di quello dell' acqua nei sanitari. L'agent controlla che la temperatura ambiente reale sia uguale a quella desiderata (impostata cioè dall'utente). Se così non fosse il riscaldamento resterà acceso fino al raggiungimento di essa.
Si potrà inoltre verificare il funzionamento di ogni singolo termosifone installato controllandone lo stato e la temperatura all'interno di esso.
E' presente anche una tabella creata per la programmazione automatica giornaliera e settimanale delle attività di riscaldamento, con la quale si potranno impostare il giorno, l'ora di accensione e di spegnimento del riscaldamento, così come la temperatura ambiente voluta. L'agent verificherà sempre la temperatura ambiente reale e il riscaldamento rimarrà attivo fino al raggiungimento della temperatura impostata (così come nell'impostazione manuale).
Ciò consente un più intelligente consumo energetico, visto che l'attività di riscaldamento si spegne automaticamente se all'interno dell'abitazione c'è la temperatura desiderata .

I consumi energetici medi di luce, acqua e gas potranno essere monitorati grazie alla tabella consumptionBoiler.

Un'altra interessante funzione è quella che permette di evitare il congelamento dell'acqua all'interno dell' impianto. Infatti, una volta impostata questa funzione, se la temperatura ambiente scendesse al di sotto del valore di soglia *tempMinAmbThres* scatterebbe la trap *antiFreeze* che azionerebbe la funzione antigelo. Funzione utile soprattutto qualora l'appartamento resti disabitato per brevi o lunghi periodi di tempo durante i quali si possono registrare rigide temperature.

La gestione degli eventuali malfunzionamenti o cambi di stato è stata realizzata grazie al controllo di tutti quelli che sono i valori di soglia minimi e massimi tollerati dall'intero impianto. Saranno controllate la pressione, le temperature e il livello dell'acqua dell'intero impianto; così come saranno generate delle trap per ogni cambiamento di stato o malfunzionamento riscontrato.

Per quanto riguarda l'aspetto della sicurezza, è stata dichiarata una variabile (*accessPassword*) da immettere per accedere al sistema. L'agent controllerà se questa password immessa è quella giusta per accedere ai servizi forniti.



esempio di comando remoto

2.1 DESCRIZIONE DELLE VARIABILI

1. **idBoiler**: racchiude i dati identificativi della caldaia ed è composto dalla sequenza dei seguenti valori:

    1.1 **codChassis**: indica il numero di telaio della caldaia
    1.2 **manAndMod**: indica marca e modello della caldaia
    1.3 **productionDate**: indica la data di produzione della caldaia
    1.4 **owner**. indica il nome del proprietario della caldaia

2. **consumptionBoile**r: tabella riguardante i consumi energetici medi della caldaia e comprende questa sequenza di valori:

    2.1 **monthlyConGas**: indica il consumo mensile medio di gas
    2.2 **monthlyConElecPow**: indica il consumo mensile medio di energia elettrica
    2.3 **monthlyConWat**: indica il consumo mensile medio di acqua

3. **accessPassword**: indica la parola chiave con cui verrà confrontata la stringa immessa dall'amministratore del sistema
    per l'accesso ad esso.

4. **boilerState**: indica lo stato di funzionamento corrente della caldaia (true= acceso, false=spento).

5. **heatingState**: indica lo stato corrente del riscaldamento (true= acceso, false=spento).

6. **sanWaterState**: indica lo stato corrente dell'acqua dei sanitari (true= acceso, false=spento).

7. **sanWaterTemperature**: indica la temperatura corrente dell'acqua dei sanitari.

8. **setSanWaterTemp**: indica la temperatura dell'acqua dei sanitari impostata dall'utente.

9. **radiatorWaterTemp**: indica la temperatura dell'acqua dei radiatori.

10. **setRadiatorWaterTemp**: indica la temperatura dei radiatori impostata dall'utente.

11. **ambientTemp**: è il termometro che indica la temperatura ambiente effettiva.

12. **wishedAmbientTemp**: indica la temperatura ambiente settata dall'utente che vorrebbe avere al momento in casa.

13. **currDateAndTime**: indica la data e l'ora corrente.

14. **heatingPressure**: indica il valore di pressione corrente nell'impianto di riscaldamento.

15. **sanPressure**: indica il valore di pressione corrente dell'acqua dei sanitari.

16. **watBoilLev**: indica il livello dell'acqua all'interno della caldaia.

17. **progAutoState**: indica se è attiva o meno la programmazione automatica del riscaldamento(true=accesa,
false=spenta).

18. **progAuto**: tabella che contiene una sequenza di oggetti riguardanti la programmazione automatica del riscaldamento:

    18.1 **dayOfWeek**: indica i giorni della settimana in cui settare le seguenti variabili (è l'indice della tabella)
    18.2 **heaStartTime**: indica l'ora di accensione per quel giorno della settimana
    18.3 **heaStopTime**: indica l'ora di spegnimento per quel giorno della settimana
    18.4 **wishAmbTemp**: indica la temperatura ambiente settata dall'utente per l'intervallo di tempo heaStartTime – heaStopTime per il giorno precedentemente selezionato(con dayOfWeek).

19. **singleRadContr**: tabella per il controllo dei singoli radiatori costituita dalla seguente sequenza di oggetti:

    19.1 **radiator**: indica il radiatore che si vuole controllare
    19.2 **radiatorState**: indica se il radiatore in questione è attivo o no
    19.3 **tempRadiator**: indica la temperatura corrente del radiatore

2.1 DESCRIZIONE DEI VALORI DI SOGLIA

1. **sanMaxTempThres**: indica la massima temperatura supportata per l'acqua dei sanitari.
2. **sanMinTempThres**: indica la temperatura minima possibile per l'acqua dei sanitari.
3. **heatMaxTempThres**: indica la temperatura massima possibile per il riscaldamento.
4. **heatMinTempThres**: indica la temperatura minima possibile per il riscaldamento.
5. **sanMaxPressThres**: indica la massima pressione sopportata dall'impianto per l'acqua dei sanitari.
6. **sanMinPressThres**: indica il valore di pressione minimo necessario per l'acqua dei sanitari.
7. **heatMaxPressThres**: indica il valore di pressione massimo sopportato per il riscaldamento.
8. **heatMinPressThres**: indica il valore di pressione minimo necessario per il riscaldamento.
9. **watMaxLevThres**: indica il limite massimo per il livello dell'acqua all'interno della caldaia.
10. **watMinLevThres**: indica il valore minimo del livello dell'acqua necessario all'interno della caldaia.
11. **tempMinAmbThres**: indica la temperatura ambiente sotto la quale viene generata la trap per la funzione antigelo.

2.3. DESCRIZIONE DELLE TRAP

1. **chanBoilerState**: trap generata quando cambia lo stato di funzionamento della caldaia (da on a off e viceversa).
2. **chanRadiatorState**: trap generata quando cambia lo stato di funzionamento di un radiatore.
3. **chanHeatingState**: trap generata quando cambia lo stato di attività dell'impianto di riscaldamento.
4. **chanSanWaterState**: trap generata se cambia lo stato d'uso dell'acqua sanitari.
5. **chanAutoProgState**: trap generata se cambia lo stato da funzionamento manuale ad automatico (o viceversa) del riscaldamento.
6. **malTempSanMax**: trap generata se la temperatura dell'acqua deii sanitari supera i valori di soglia(sanMaxTempThres).

7. **malTempSanMin**: trap generata se la temperatura dell'acqua dei sanitari si abbassa sotto la soglia minima.

8. **malTempHeatMax**: trap generata quando la temperatura del radiatore supera i valori massimi consentiti.

9. **malTempHeatMin**: trap generata quando la temperatura del radiatore si abbassa sotto il valore minimo.

10. **malPressSanMax**: trap generata quando la pressione dell'acqua dei sanitari supera il valore massimo.

11. **malPressSanMin**: trap generata quando la pressione dell'acqua dei sanitari si abbassa sotto il valore minimo.

12. **malPressHeatMin**: trap generata quando la pressione dell'impianto riscaldamento supera la soglia massima.

13. **malPressHeatMin**: trap generata quando la pressione dell'impianto riscaldamento si abbassa sotto la soglia minima.

14. **malLevWatMax**: trap generata quando il livello dell'acqua nella caldaia supera il valore massimo.

15. **malLevWatMin**: trap generata quando il livello dell'acqua nella caldaia si abbassa oltre il valore minimo.

16. **antiFreezeTrap**: trap generata quando la temperatura ambiente è minore della temperatura ambiente minima consentita (tempMinAmbThres) e quando lo stato del riscaldamento è off. Quando questa trap è generata scatta la funzione antigelo, cioè il riscaldamento si accende fino a quando la temperatura ambiente non ritorna al di sopra o uguale al valore minimo di temperatura ambiente consentita (tempMinAmbThres).

Dettaglio di una caldaia: barometro

3. IL MIB

```
BOILER-MIB DEFINITIONS::=BEGIN

IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE,
        NOTIFICATION-TYPE,
        Unsigned32, Gauge32                          FROM SNMPv2-SMI
        DisplayString                                FROM SNMPv2-TC;

boilerMIB MODULE-IDENTITY
          LAST-UPDATED          "200306091204Z"
          ORGANIZATION          "BERNARDI & ROSSI S.p.A."
          CONTACT-INFO
          "
             Bernardi Andrea
              e-mail:  andrea2031@interfree.it

             Rossi Roberto
              e-mail: robros@comeg.it
          "
          DESCRIPTION           "MIB module for management of gas murals boilers"
::={private75}

boilerObject OBJECT  IDENTIFIER
::={boilerMIB 2}
boilerThreshold OBJECT  IDENTIFIER
::={boilerMIB 3}
boilerTrap OBJECT  IDENTIFIER
::={boilerMIB 4}
```

--OBJECTS' DESCRIPTION

```
idBoilerTable  OBJECT-TYPE
                SYNTAX                  SEQUENCE OF IdBoiler
                MAX-ACCESS              not-accessible
                STATUS          current
                DESCRIPTION             "Description of general features of the boiler"
 ::={boilerObject 1}


idBoiler  OBJECT-TYPE
                SYNTAX                  IdBoiler
                MAX-ACCESS              not-accessible
                STATUS          current
                DESCRIPTION             "An interface entry containing objects  regarding
                                         the identification of  the boiler"
                INDEX                   { codChassis }
 ::={ idBoilerTable 1}


IdBoiler ::=
        SEQUENCE  {
                codChassis
                        DisplayString,
```

```
                manAndMod
                        DisplayString,
                productionDate
                        Unsigned32,
                owner
                        DisplayString
                         }

codChassis    OBJECT-TYPE
              SYNTAX                DisplayString
              MAX-ACCESS            read-only
              STATUS                current
              DESCRIPTION           "It indicates the code of the chassis"
 ::={ idBoiler 1}

manAndMod  OBJECT-TYPE
              SYNTAX                DisplayString
              MAX-ACCESS            read-only
              STATUS          current
              DESCRIPTION           "It indicates marks and model of the boiler"
 ::={ idBoiler 2}

productionDate  OBJECT-TYPE
              SYNTAX                Unsigned32
              MAX-ACCESS            read-only
              STATUS          current
              DESCRIPTION           "It indicates the date of production of the boiler"
 ::={ idBoiler 3}

owner  OBJECT-TYPE
              SYNTAX                DisplayString
              MAX-ACCESS            read-write
              STATUS          current
              DESCRIPTION           "It indicates the owner of the boiler"
 ::={ idBoiler 4}


consumptionBoilerTable  OBJECT-TYPE
              SYNTAX                SEQUENCE OF ConsumptionBoiler
              MAX-ACCESS            not-accessible
              STATUS          current
              DESCRIPTION           "Sequence of values about boiler's
                                    consumptions"
 ::={boilerObject 2}

consumptionBoiler  OBJECT-TYPE
              SYNTAX                ConsumptionBoiler
              MAX-ACCESS            not-accessible
              STATUS          current
              DESCRIPTION           "An interface entry containing objects  regarding
                                    the consumptions of  the boiler"
              INDEX                 { monthlyConGas }
 ::={ consumptionBoilerTable 1}

ConsumptionBoiler ::=
```

```
        SEQUENCE {
                monthlyConGas
                        Unsigned32,
                monthlyConElecPow
                        Unsigned32,
                monthlyConWat
                        Unsigned32
                 }


monthlyConGas  OBJECT-TYPE
                SYNTAX              Unsigned32
                MAX-ACCESS          read-only
                STATUS          current
                DESCRIPTION         "It indicates the monthly gas consumption  of the
                                    boiler"
 ::={ consumptionBoiler 1 }


monthlyConElecPow  OBJECT-TYPE
                SYNTAX              Unsigned32
                MAX-ACCESS          read-only
                STATUS          current
                DESCRIPTION         "It indicates the monthly Electric power
                                consumption  of the boiler"
 ::={ consumptionBoiler 2 }


monthlyConWat  OBJECT-TYPE
                SYNTAX              Unsigned32
                MAX-ACCESS          read-only
                STATUS          current
                DESCRIPTION         "It indicates the monthly water's
                                consumption  of the boiler"
 ::={ consumptionBoiler 3 }


accessPassword OBJECT-TYPE
                SYNTAX              DisplayString
                MAX-ACCESS          read-write
                STATUS          current
                DESCRIPTION         "It indicates the access' password to entry in the
                                system, the agent checks if the password
                                    introduced by the user is correct"
 ::={ boilerObject 3}




boilerState  OBJECT-TYPE
                SYNTAX              Boolean
                MAX-ACCESS          read-write
                STATUS          current
                DESCRIPTION         "Boiler's state: true is on, false is off"

::={ boilerObject 4}


heatingState  OBJECT-TYPE
                SYNTAX              Boolean
                MAX-ACCESS          read-write
                STATUS          current
```

DESCRIPTION "Heating's state: true is on, false is off"

::={ boilerObject 5}

sanWaterState ***OBJECT-TYPE***
SYNTAX Boolean
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Sanitary's water state: true is on (someone is using it) , false is off (nobody is using it)"

::={ boilerObject 6}

sanWaterTemperature ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the current temperature of the sanitary's water"

::={ boilerObject 7}

setSanWaterTemp ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "It indicates the user's selected temperature of the sanitary's water"

::={ boilerObject 8}

radiatorWaterTemp ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the current temperature of the radiator's water"

::={ boilerObject 9}

setRadiatorWaterTemp ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "It indicates the user's selected temperature of the radiator's water"

::={ boilerObject 10}

ambientTemp ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It is the thermometer that indicates the current

ambient's temperature where the boiler is
installed"

::={ boilerObject 11}

wishedAmbientTemp ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "It is the thermometer that indicates the current ambient's temperature selected by user. If the ambientTemp variable is minor than this variable, the heating is on until the values of the two variables are equals"

::={ boilerObject 12}

currDateAndTime ***OBJECT-TYPE***
SYNTAX Unsigned32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "It indicates the current date and time"

::={ boilerObject 13}

heatingPressure ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the current boiler's pressure for the heating's system"

::={ boilerObject 14}

sanPressure ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the current boiler's pressure for the sanitary's system"

::={ boilerObject 15}

watBoilLev ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the current boiler's level of water inside it"

::={ boilerObject 16}

progAutoState ***OBJECT-TYPE***
SYNTAX Boolean
MAX-ACCESS read-write
STATUS current

```
                DESCRIPTION         "It indicates if the automatic programming of
                                    heating is on or off; the user can decides if to use
                                    the automatic or manual mode; when this
                                    variable is off, it is selected the manual mode"

::={ boilerObject 17}


progAutoTable  OBJECT-TYPE
                SYNTAX              SEQUENCE OF ProgAuto
                MAX-ACCESS          not-accessible
                STATUS          current
                DESCRIPTION         "Table for the automatic programming of the
                                    heating "
 ::={boilerObject 18}


progAuto  OBJECT-TYPE
                SYNTAX              ProgAuto
                MAX-ACCESS          not-accessible
                STATUS          current
                DESCRIPTION         "An interface entry containing objects  regarding
                                    the automatic programming of the heating"
                INDEX               { dayOfWeek }
 ::={ progAutoTable 1}


ProgAuto ::=
        SEQUENCE  {
                dayOfWeek
                        DisplayString,
                heaStartTime
                        Unsigned32,
                heaStopTime
                        Unsigned32,
                wishAmbTemp
                        Gauge32
                         }


dayOfWeek  OBJECT-TYPE
                SYNTAX              DisplayString
                MAX-ACCESS          read-only
                STATUS          current
                DESCRIPTION         "It indicates the days of the week, in everyone of
                                    which is set the starting and the stopping time of
                                    the heating"
 ::={ progAuto 1}




heaStartTime  OBJECT-TYPE
                SYNTAX              Unsigned32
                MAX-ACCESS          read-write
                STATUS          current
                DESCRIPTION         "It indicates the selected starting time of the
                                    heating in the corresponding day of week"
```

::={ progAuto 2}

heaStopTime ***OBJECT-TYPE***
SYNTAX Unsigned32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "It indicates the selected stopping time of the heating in the corresponding day of week"

::={ progAuto 3}

wishAmbTemp ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "It is the thermometer that indicates the current ambient's temperature selected by user. If the ambientTemp variable is minor than this variable, the heating is on until the values of the two variables are equals, or however is on during the selected time interval of starting and stopping of the heating"

::={ progAuto 4}

singleRadContrTable ***OBJECT-TYPE***
SYNTAX SEQUENCE OF SingleRadContr
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Table to control the singles radiators"

::={boilerObject 19}

singleRadContr ***OBJECT-TYPE***
SYNTAX SingleRadContr
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "An interface entry containing objects regarding the control of singles radiators"
INDEX { radiator }

::={ singleRadContrTable 1}

SingleRadContr ::=
SEQUENCE {
radiator
DisplayString,
radiatorState
Boolean,
tempRadiator
Gauge32
}

radiator ***OBJECT-TYPE***
SYNTAX DisplayString
MAX-ACCESS read-write
STATUS current
DESCRIPTION "It indicates the radiator to control"

::={ singleRadContr 1 }

radiatorState ***OBJECT-TYPE***
SYNTAX Boolean
MAX-ACCESS read-write
STATUS current
DESCRIPTION "It indicates if the radiator is on(true) or
off(false), the user can power on or turn off
every single radiator"

::={ singleRadContr 2 }

tempRadiator ***OBJECT-TYPE***
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the current radiator's temperature"

::={ singleRadContr 3 }

--THRESHOLD'S DESCRIPTION

sanMaxTempThres ***OBJECT-TYPE***
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the max. possible temperature of the
sanitary's water"

::={ boilerThreshold 1}

sanMinTempThres ***OBJECT-TYPE***
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the min. possible temperature of the
sanitary's water"

::={ boilerThreshold 2}

heatMaxTempThres ***OBJECT-TYPE***
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the max. possible temperature of the

heating"

::={ boilerThreshold 3}

heatMinTempThres ***OBJECT-TYPE***

SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	"It indicates the min. possible temperature of the heating"

::={ boilerThreshold 4}

sanMaxPressThres ***OBJECT-TYPE***

SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	"It indicates the max. possible pressure of the sanitary's system"

::={ boilerThreshold 5}

sanMinPressThres ***OBJECT-TYPE***

SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	"It indicates the min. possible pressure of the sanitary's system"

::={ boilerThreshold 6}

heatMaxPressThres ***OBJECT-TYPE***

SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	"It indicates the max. possible pressure of the heating's system"

::={ boilerThreshold 7}

heatMinPressThres ***OBJECT-TYPE***

SYNTAX	Unsigned32
MAX-ACCESS	read-only
STATUS	current
DESCRIPTION	"It indicates the min. possible pressure of the heating's system"

::={ boilerThreshold 8}

watMaxLevThres ***OBJECT-TYPE***

SYNTAX	Unsigned32

MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the max. possible level of the water inside the boiler"

::={ boilerThreshold 9}

watMinLevThres ***OBJECT-TYPE***
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the min. possible level of the water inside the boiler"

::={ boilerThreshold 10}

tempMinAmbThres ***OBJECT-TYPE***
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It indicates the min. possible ambient's temperature. It is useful to activate the antifreeze function "

::={ boilerThreshold 11}

--TRAP'S DESCRIPTION

chanBoilerState ***NOTIFICATION-TYPE***
OBJECTS {boilerState}
STATUS current
DESCRIPTION "It is generated when the state of the boiler is changed (from on to off or vice versa)"
::={boilerTrap 1}

chanRadiatorState ***NOTIFICATION-TYPE***
OBJECTS {radiatorState}
STATUS current
DESCRIPTION "It is generated when the state of a radiator is changed (from on to off or vice versa)"
::={boilerTrap 2}

chanHeatingState ***NOTIFICATION-TYPE***
OBJECTS {heatingState}
STATUS current
DESCRIPTION "It is generated when the state of the heating is changed (from on to off or vice versa)"
::={boilerTrap 3}

```
chanSanWaterState NOTIFICATION-TYPE
                    OBJECTS          {sanWaterState}
                    STATUS           current
                    DESCRIPTION      "It is generated when the state of the use
                                     of the Sanitary's water is changed (from
                                     on to off or vice versa)"
::={boilerTrap 4}


chanAutoProgState NOTIFICATION-TYPE
                    OBJECTS          {progAutoState}
                    STATUS           current
                    DESCRIPTION      "It is generated when the state of the
                                     automatic programming (of the heating)
                                     is changed (from on to off or vice versa)"
 ::={boilerTrap 5}


malTempSanMax NOTIFICATION-TYPE
                    OBJECTS          { sanWaterTemperature }
                    STATUS           current
                    DESCRIPTION  "It is generated when the sanitary's water
                                   temperature exceeds the threshold of the
                                   max. temperature possible for the
                                   sanitary's water (sanMaxTempThres)"
::={boilerTrap 6}


malTempSanMin NOTIFICATION-TYPE
                    OBJECTS          { sanWaterTemperature }
                    STATUS           current
                    DESCRIPTION      "It is generated when the sanitary's water
                                   temperature exceeds the threshold of the
                                   min. temperature possible for the
                                   sanitary's water (sanMinTempThres)"
::={boilerTrap 7}


malTempHeatMax NOTIFICATION-TYPE
                    OBJECTS          { radiatorWaterTemp }
                    STATUS           current
                    DESCRIPTION      "It is generated when the radiator's water
                                     temperature exceeds the threshold of the
                                     max. temperature possible for the
                                     radiator's water (heatMaxTempThres)"
::={boilerTrap 8}


malTempHeatMin NOTIFICATION-TYPE
                    OBJECTS          { radiatorWaterTemp }
                    STATUS           current
                    DESCRIPTION      "It is generated when the radiator's water
                                     temperature exceeds the threshold of the
                                   min. temperature possible for the
```

radiator's water (heatMinTempThres)"
::={boilerTrap 9}

malPressSanMax ***NOTIFICATION-TYPE***
OBJECTS { sanPressure }
STATUS current
DESCRIPTION "It is generated when the sanitary's pressure exceeds the threshold of the max. pressure possible for the sanitary's system (sanMaxPressThres)"
::={boilerTrap 10}

malPressSanMin ***NOTIFICATION-TYPE***
OBJECTS { sanPressure }
STATUS current
DESCRIPTION "It is generated when the sanitary's pressure exceeds the threshold of the min. pressure possible for the sanitary's system (sanMinPressThres)"
::={boilerTrap 11}

malPressHeatMax ***NOTIFICATION-TYPE***
OBJECTS { heatingPressure }
STATUS current
DESCRIPTION "It is generated when the heating's pressure exceeds the threshold of the max. pressure possible for the heating's system (heatMaxPressThres)"
::={boilerTrap 12}

malPressHeatMin ***NOTIFICATION-TYPE***
OBJECTS { heatingPressure }
STATUS current
DESCRIPTION "It is generated when the heating's pressure exceeds the threshold of the min. pressure possible for the heating's system (heatMinPressThres)"
::={boilerTrap 13}

malLevWatMax ***NOTIFICATION-TYPE***
OBJECTS { watBoilLev }
STATUS current
DESCRIPTION "It is generated when the boiler's water level exceeds the threshold of the max. level possible for the water inside the boiler (watMaxLevThres)"
::={boilerTrap 14}

malLevWatMin ***NOTIFICATION-TYPE***
OBJECTS { watBoilLev }
STATUS current
DESCRIPTION "It is generated when the boiler's water level exceeds the threshold of the

```
                                        min. level possible for the water inside
                                        the boiler (watMinLevThres)"
::={boilerTrap 15}

antifreezeTrap NOTIFICATION-TYPE
                        OBJECTS         { ambientTemp }
                        STATUS          current
                        DESCRIPTION     "It is generated when the ambient's
                                        temperature exceeds the threshold of the
                                        min. ambient's temperature
                                        (tempMinAmbThres) and the heating's
                                        state is off. When this trap is sent, the
                                        heating is activated (so the heating's state
                                        becomes on) until the ambient's
                                        temperature returns more or equal than
                                        ' tempMinAmbThres ' value"
::={boilerTrap 16}

END.
```

4. SVILUPPI FUTURI

In generale gli sviluppi futuri, in questo campo, sono orientati verso l'estensione delle funzionalità di gestione da remoto di tutta un'intera abitazione, quindi verso il controllo a distanza di ogni elettrodomestico, porte, allarmi, finestre, ecc. Sfruttando poi la diffusione sempre più grande delle tecnologie associate al web, la gestione di intere abitazioni può avvenire anche trovandosi a migliaia di chilometri di distanza da esse.
Per quel che riguarda poi nello specifico l'oggetto del nostro progetto, possiamo pensare ad un miglioramento nella funzione della programmazione automatica del riscaldamento, aggiungendo la possibilità di utilizzare più di un intervallo di tempo nell'arco di uno stesso giorno per l'accensione e lo spegnimento automatico del riscaldamento.
Un'altra interessante funzione, che possiamo aggiungere al nostro progetto, e' quella cosiddetta del "programma vacanze". Essa prevede il mantenimento di una bassa temperatura costante quando si lascia, per esempio, l'abitazione per le vacanze, o comunque quando ci si assenta per brevi periodi durante la giornata. Si potrà poi, grazie a questa funzione, impostare per esempio una a scelta fra più modalità, come accendere il riscaldamento alla mezzanotte o ad un'ora precisa del giorno del rientro, o ancora accenderlo più tardi il giorno stesso in cui si lascia l'abitazione, e così via.

5. CONCLUSIONI

La stesura di questo MIB ha richiesto l'analisi di tutte quelle parti che contribuiscono al buon funzionamento e al controllo di una caldaia murale a gas.
Lo scopo di questo progetto è prettamente didattico e non ha la pretesa di essere un documento completo per l' intera gestione di una caldaia murale a gas. E' piuttosto un semplice esempio di come è possibile gestire, mediante il protocollo SNMP, uno dei più comuni e importanti elettrodomestici di utilizzo quotidiano che si possono trovare in una casa. Possiamo quindi affermare che questo MIB è in pieno stile SNMP, che ha fatto della semplicità il suo punto di forza.

6. RIFERIMENTI

J. Schönwälder, L.Deri “Sistemi di elaborazione dell'informazione: Gestione di Rete” v. 1.3

L ‘ intero MIB è stato testato al sito: http://www.simpleweb.org/ietf/mibs/validate/

RFC 1157 : definizione del protocollo SNMP

RFC 1213: definizione di MIB-II

ESSENTIAL SNMP, Charter 2: A closet look at SNMP By Douglas Mauro & Kevin Schmidt , ed. O’REILLY

Manuale per l’installazione ed il funzionamento di caldaia murale a gas “Ferella free f 24 mel” (ditta produttrice: FER industrie)

Progetto interamente ideato e sviluppato da:
Andrea Bernardi e Roberto Rossi
(Un ringraziamento speciale al fotografo ufficiale del progetto: Gianluca Romoli)