

# Definizione di un MIB per il monitoraggio di una moto da corsa



*Antonio Bartolini*  
N. Mat. 271056  
[robbynhub@gmail.com](mailto:robbynhub@gmail.com)

*Corso di Complementi di Gestione di Reti - Antonio Bartolini*  
Questo documento e' pubblicato sotto licenza GPL2 ([www.gnu.org](http://www.gnu.org)) in ogni sua parte.

## Indice

1-Introduzione .....	pag 3
2-Scelte implementative .....	pag 4
3-Descrizione del MIB .....	pag 6
3.1-Descrizione variabili utilizzate .....	pag 6
3.2-Descrizione tabelle.....	pag 7
3.3-Descrizione trap.....	pag 8
3.4-Descrizione trapthreshold.....	pag 9
4-Definizione MIB .....	pag 10
5-Conclusioni .....	pag 20
5.1-Sviluppi futuri .....	pag 20
6-Riferimenti .....	pag 20

## Introduzione

Durante una corsa motociclistica e' indispensabile tenere sempre sotto controllo le condizioni della moto. I meccanici devono sempre sapere cosa gli aspetta durante la preparazione dei pit-stop e le informazioni trasmesse dal pilota sono spesso insufficienti per mantenere un controllo totale della moto.

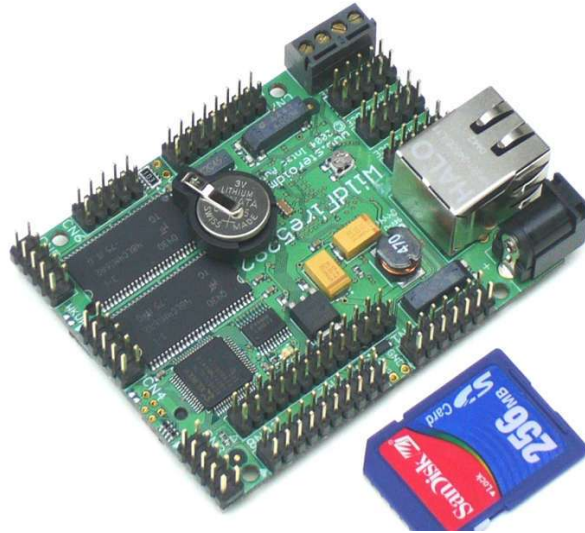
Per risolvere tale problema ho voluto descrivere una possibile implementazione di un MIB che mantenga sotto controllo tutti gli aspetti della moto senza risultare troppo invadente.



Un calo di pressione nel circuito dell' olio o una temperatura dei gas di scarico troppo alta puo' addirittura pregiudicare la vita del pilota se non controllata in tempo.

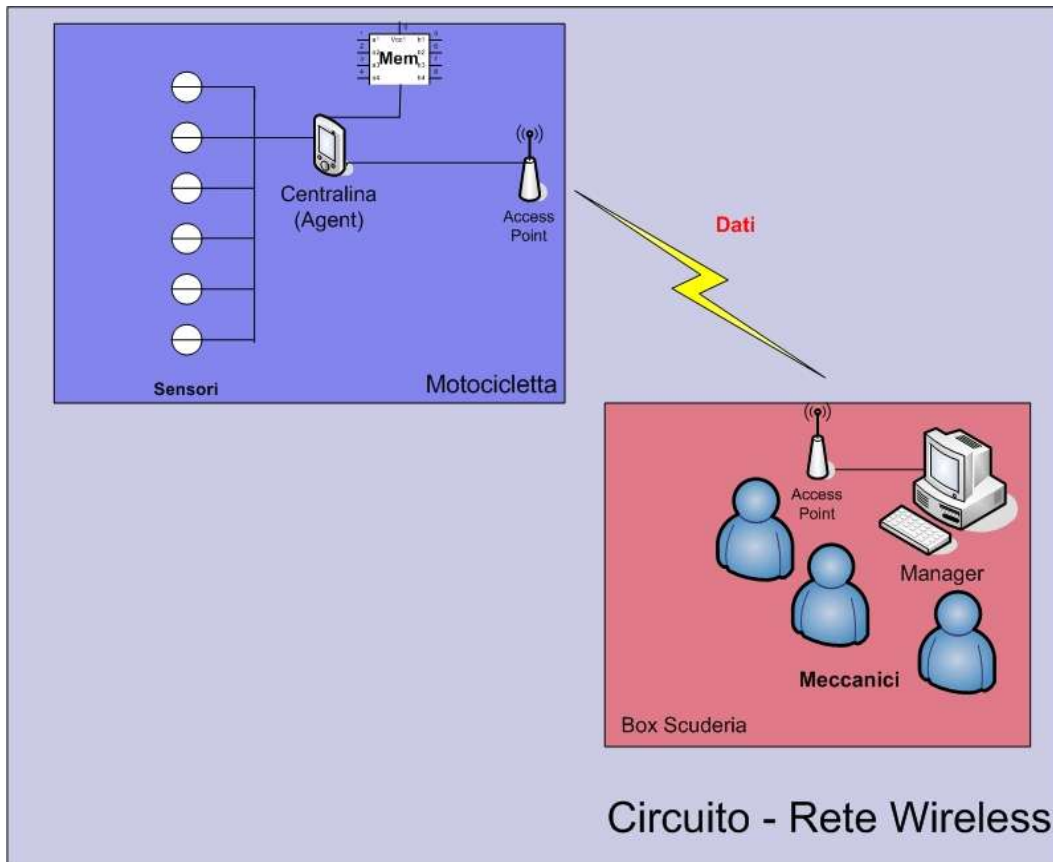
## Scelte Implementative

Per poter mantenere il peso di tutto il sistema piu' leggero possibile, ho deciso di utilizzare una centralina "WildFire 5282" prodotta dalla Intec Automation Inc. (<http://www.steroidmicros.com>).



Questa centralina e' stata progettata esattamente per scopi come questo. Ha in dotazione un processore a 64 Mhz FreeScale ColdFire2 integrato con 4 Mb di spazio e ben 16 Mb di SDRAM veloce per l' esecuzione del software. Essa supporta fino a 16 sensori digitali e 8 sensori analogici con lettura a 10 bit e 140 Mhz di frequenza. Una delle sue particolarita' piu' importanti, c'e' la possibilita' di utilizzare le sue 7 Porte Interrupt per poter gestire tempestivamente le trap in caso di malfunzionamento meccanico del motore. Questa scheda possiede anche una porta Ethernet integrata ma, per la trasmissione dei dati, si prevede di utilizzare un modulo 802.11 al posto del lettore di SD cards. Per alimentarla e' sufficiente una qualsiasi corrente continua da 6 a 24 volts ma si prevede di utilizzare un filtro per eliminare i picchi di tensione generati all' accensione della moto. Le sue dimensioni contenute (7 x 4 cm.) la rendono perfetta per il suo scopo. La sua programmazione avviene mediante linguaggio C con un apposito compilatore offerto in dotazione dalla Intec assieme alla scheda.

Il MIB sviluppato, verra' implementato su questo tipo di scheda e i dati verranno trasmessi direttamente all' Access Point presente nel box corse dove verranno analizzati in tempo reale dai meccanici.



Per prevenire eventuali perdite di segnale o power failure, una copia dei dati verra' mantenuta sulla scheda che funge cosi' anche da scatola nera.

La centralina monitorizzera' tutti quei parametri indispensabili ai meccanici e agli ingegneri per tenere sempre sotto controllo la moto. Le trap vengono inviate direttamente in base ai valori presenti della tabella delle soglie e sono modificabili tramite snmp.

# Descrizione del MIB

## 3.1-Descrizione delle variabili.

**NOTA: Tutte le seguenti variabili sono accessibili solo in modalita' readonly eccetto dove specificato.**

### *lapCount Counter32*

- questa variabile identifica il giro attuale contato a partire dall' accensione della moto.

### *fuelLevel Gauge32*

- questa variabile indica la quantita'di benzina espressa in litri nel serbatoio.

### *voltage Gauge32*

- questa variabile indica il livello di tensione espresso in volts dell' impianto elettrico.

### *oilLevel Gauge32*

- questa variabile indica la quantita' in litri di olio nel motore.

### *engineRPM Gauge32*

- questa variabile indica il numero di giri al minuto del motore.

### *brakeOilLevel Gauge32*

- questa variabile indica la quantita' in litri di olio per l'impianto frenante.

### *brakeOilPressure Gauge32*

- questa variabile indica la pressione dell' olio espresso in BAR.

### *brakePastilles Gauge32*

- questa variabile indica lo spessore in millimetri delle pasticche dei freni.

### *exhaustTemp Gauge32*

- questa variabile indica la temperatura dei gas di scarico espressa in gradi centigradi.

### *engineTemp Gauge32*

- questa variabile indica la temperatura del motore espressa in gradi centigradi.

### *speed Gauge32*

- questa variabile indica in chilometri orari la velocita' istantanea del veicolo.

### *averageSpeed Gauge32*

- questa variabile indica in chilometri orari la velocita' media calcolata dal momento dall' attivazione dell' agent.

### *maxSpeed Gauge32*

- questa variabile indica in chilometri orari la velocita' massima raggiunta dal momento dell' attivazione dell' agent.

### *waterTemp Gauge32*

- questa variabile indica la temperatura espressa in gradi centigradi del liquido di raffreddamento del motore.

### *waterLevel Gauge32*

- questa variabile indica il livello del liquido di raffreddamento espresso in litri.

*ampere Gauge32*

- questa variabile indica il livello di consumo elettrico espresso in Ampere.

*gearBoxTemp Gauge32*

- questa variabile indica la temperatura espressa in gradi centigradi della scatola del cambio.

*clutchTemp Gauge32*

- questa variabile indica la temperatura espressa in gradi centigradi della scatola della frizione.

*engineHP Gauge32*

- questa variabile indica la potenza istantanea generata dal motore espressa in cavalli vapore.

*brake1Temp Gauge32*

- questa variabile indica la temperatura espressa in gradi centigradi dei dischi frenanti anteriori.

*brake2Temp Gauge32*

- questa variabile indica la temperatura espressa in gradi centigradi dei dischi frenanti posteriori.

*pilotName DisplayString [read-write]*

- questa variabile indica il nome del pilota che guida il veicolo.

*model DisplayString [read-write]*

- questa variabile indica marca e modello del veicolo.

### 3.3-Descrizione tabelle

**NOTA: Tutte le seguenti variabili sono accessibili solo in modalita' readonly.**

Tabella: *lapTable*

Questa tabella contiene le informazioni riguardanti il comportamento della moto per ogni giro di pista.

*lap Counter32*

- questa variabile identifica il giro di pista.

*usedFuel Integer32*

- questa variabile indica la quantita'di benzina consumata durante il giro.

*usedOil Integer32*

- questa variabile indica la quantita'di olio consumata durante il giro.

*averageSpeedLap Integer32*

- questa variabile indica la velocita' media durante il giro.

*averagePower Integer32*

- questa variabile indica l' energia media generata dal motore durante il giro espressa in Kjoule su secondo.

*gap Integer32*

- questa variabile indica la differenza di tempo rispetto al giro precedente espresso in centesimi di secondo.

### 3.4-Descrizione trap

#### *lowFuel*

- generata quando la quantità di benzina scende sotto una soglia minima predefinita nella sezione TrapThresholds.

#### *highVoltage*

- generata quando la tensione dell' impianto elettrico supera una soglia predefinita nella sezione TrapThresholds.

#### *lowVoltage*

- generata quando la tensione dell' impianto elettrico scende sotto una soglia predefinita nella sezione TrapThresholds.

#### *lowOilLevel*

- generata quando la quantità di olio nel motore scende sotto una soglia predefinita nella sezione TrapThresholds.

#### *outOfRPM*

- generata quando il numero di giri del motore supera la soglia massima predefinita nella sezione TrapThresholds.

#### *lowBrakeOil*

- generata quando il livello dell' olio dell' impianto frenante scende sotto la soglia minima predefinita nella sezione TrapThresholds.

#### *lowBrakePressure*

- generata quando la pressione dell' olio nell' impianto frenante scende sotto una soglia minima predefinita nella sezione TrapThresholds.

#### *usedPastillesWarning*

- generata quando la pressione dell' olio nell' impianto frenante scende sotto una soglia minima predefinita nella sezione TrapThresholds.

#### *highExhaustTemp*

- generata quando la temperatura dei gas di scarico sale sopra una soglia prestabilita nella sezione TrapThresholds.

#### *highEngineTemp*

- generata quando la temperatura della testata sale sopra una soglia prestabilita nella sezione TrapThresholds.

#### *highWaterTemp*

- generata quando la temperatura del liquido di raffreddamento sale sopra una soglia prestabilita nella sezione TrapThresholds.

#### *lowWaterLevel*

- generata quando il livello del liquido di raffreddamento e' sotto la soglia di guardia prestabilita nella sezione TrapThresholds.

#### *highGearBoxTemp*

- generata quando la temperatura della scatola del cambio supera una soglia massima prestabilita nella sezione TrapThresholds.

#### *highClutchTemp*

- generata quando la temperatura dei dischi della frizione superano una soglia massima



prestabilita nella sezione TrapThresholds.

*highBrake1Temp*

- generata quando la temperatura del disco frenante anteriore supera una soglia massima prestabilita nella sezione TrapThresholds.

*highBrake2Temp*

- generata quando la temperatura del disco frenante anteriore supera una soglia massima prestabilita nella sezione TrapThresholds.

### 3.4 - Descrizione TrapThresholds.

**NOTA: Tutte le seguenti variabili sono accessibili in modalita readwrite. E servono per definire le soglie massime/minime per l' invio delle trap.**

*lowFuelThreshold* [Read-Write] Integer32

*highVoltageThreshold* [Read-Write] Integer32

*lowVoltageThreshold* [Read-Write] Integer32

*lowOilLevelThreshold* [Read-Write] Integer32

*outOfRPMThreshold* [Read-Write] Integer32

*lowBrakeOilThreshold* [Read-Write] Integer32

*lowBrakePressureThreshold* [Read-Write] Integer32

*usedPastillesThreshold* [Read-Write] Integer32

*highExhaustTempThreshold* [Read-Write] Integer32

*highEngineTempThreshold* [Read-Write] Integer32

*highWaterTempThreshold* [Read-Write] Integer32

*lowWaterLevelThreshold* [Read-Write] Integer32

*highGearBoxTempThreshold* [Read-Write] Integer32

*highClutchTempThreshold* [Read-Write] Integer32

*highBrake1TempThreshold* [Read-Write] Integer32

*highBrake2TempThreshold* [Read-Write] Integer32

## Definizione MIB

```
MOTOMIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, Gauge32,
    Unsigned32, Counter32, Integer32 FROM SNMPv2-SMI
    enterprises FROM RFC1155-SMI
    DisplayString FROM SNMPv2-TC;

motoMIB MODULE-IDENTITY
    LAST-UPDATED "200505231616Z"
    ORGANIZATION "DB-SW"
    CONTACT-INFO
        "Antonio Bartolini
        e-mail: bartolin@cli.di.unipi.it"
    DESCRIPTION "Modulo MIB per il monitoraggio di motocicli che
    partecipano a gare su pista"
    ::= { enterprises 99 }

moto-values OBJECT IDENTIFIER
    ::= { motoMIB 1 }

moto-threshold OBJECT IDENTIFIER
    ::= { motoMIB 2 }

moto-trap OBJECT IDENTIFIER
    ::= { motoMIB 3 }

--OBJECT DEFINITIONS

lapCount OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce il numero di giri percorsi dall' avvio
    dell' agent"
    ::= { moto-values 1 }

fuelLevel OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce la quantita' di benzina presente nel
    serbatoio espressa in litri."
    ::= { moto-values 2 }

voltage OBJECT-TYPE
    SYNTAX Gauge32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Questa variabile indica il livello di tensione
    dell' impianto elettrico espresso in Volts."
    ::= { moto-values 3 }
```

```

oilLevel OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce la quantita' di olio presente nel nel
motore espresso in litri."
    ::= {moto-values 4}

engineRPM OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce il numero di giri al minuto del motore"
    ::= {moto-values 5}

brakeOilLevel OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce la quantita' di benzina presente nel
serbatoio espressa in litri."
    ::= {moto-values 6}

brakeOilPressure OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce la pressione dell' olio dei freni
espresso in BAR"
    ::= {moto-values 7}

brakePastilles OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce lo spessore delle pasticche dei freni"
    ::= {moto-values 8}

exhaustTemp OBJECT-TYPE
    SYNTAX Gauge32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce la temperatura dei gas di scarico."
    ::= {moto-values 9}

engineTemp OBJECT-TYPE
    SYNTAX Gauge32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Definisce la temperatura della testata del
motore."
    ::= {moto-values 10}

speed OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current

```

```

DESCRIPTION "Definisce la velocita' istantanea della moto
espressa in chilometri orari"
::={moto-values 11}

averageSpeed OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Definisce la velocita' media della moto dal
momento dell' attivazione dell' agent"
::={moto-values 12}

maxSpeed OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Definisce la velocita' massima raggiunta dal mezzo
espressa in chilometri all' ora"
::={moto-values 13}

waterTemp OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Definisce la temperatura espressa in gradi
centigradi del liquido di raffreddamento del motore."
::={moto-values 14}

waterLevel OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Indica il livello del liquido di raffreddamento
espresso in litri."
::={moto-values 15}

ampere OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Indica il livello di consumo elettrico della moto
espresso in Ampere"
::={moto-values 16}

gearBoxTemp OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Definisce la temperatura espressa in gradi
centigradi della scatola del cambio"
::={moto-values 17}

clutchTemp OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Definisce la temperatura espressa in gradi

```

```

centigradi della scatola della frizione"
 ::= {moto-values 18}

engineHP OBJECT-TYPE
 SYNTAX Integer32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION "Definisce la potenza istantanea sprigionata dal
 motore espressa in cavalli motore."
 ::= {moto-values 19}

brake1Temp OBJECT-TYPE
 SYNTAX Gauge32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION "Definisce la temperatura espressa in gradi
 centigradi del freno anteriore."
 ::= {moto-values 20}

brake2Temp OBJECT-TYPE
 SYNTAX Gauge32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION "Definisce la temperatura espressa in gradi
 centigradi del freno posteriore."
 ::= {moto-values 21}

pilotName OBJECT-TYPE
 SYNTAX DisplayString
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION "Nome del pilota della moto."
 ::= {moto-values 22}

model OBJECT-TYPE
 SYNTAX DisplayString
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION "Modello identificativo della moto."
 ::= {moto-values 23}

lapTable OBJECT-TYPE
 SYNTAX SEQUENCE OF LapEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "Questa tabella contiene le informazioni
 riguardanti il comportamento della moto per ogni giro di pista."
 ::= {moto-values 24}

lapEntry OBJECT-TYPE
 SYNTAX LapEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION "Entry della tabella."
 INDEX {lap}
 ::= {lapTable 1}

```

```

LapEntry ::= SEQUENCE {
    lap Counter32,
    usedFuel Integer32,
    usedOil Integer32,
    averageSpeedLap Integer32,
    averagePower Integer32,
    gap Integer32
}

lap OBJECT-TYPE
    SYNTAX Counter32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Numero del giro di pista."
    ::= {lapEntry 1}

usedFuel OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Consumo di carburante durante l' ultimo giro. Espresso in litri."
    ::= {lapEntry 2}

usedOil OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Consumo di carburante durante l' ultimo giro. Espresso in centilitri"
    ::= {lapEntry 3}

averageSpeedLap OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Velocita' media del giro espressa in chilometri orari"
    ::= {lapEntry 4}

averagePower OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Energia media generata dal motore espressa in Kjoule su secondo"
    ::= {lapEntry 5}

gap OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION "Differenza di tempo di percorrenza rispetto al giro precedente espresso in centesimi di secondo"
    ::= {lapEntry 6}

```

--THRESHOLD DEFINITIONS

```
lowFuelThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "Definisce il numero di litri minimo presente nel
    serbatoio per l' invio della trap"
    ::= {moto-threshold 1}

highVoltageThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "Definisce il numero di Volts massimo presente nel
    circuito della moto per l' invio della trap"
    ::= {moto-threshold 2}

lowVoltageThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "Definisce il numero di Volts minimo presente nel
    circuito della moto per l' invio della trap"
    ::= {moto-threshold 3}

lowOilLevelThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "Definisce il numero di litri minimo di olio
    presente nel motore per l' invio della trap"
    ::= {moto-threshold 4}

outOfRPMThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "Definisce il numero massimo di giri al minuto del
    motore per l' invio della trap"
    ::= {moto-threshold 5}

lowBrakeOilThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "Definisce il numero di litri minimo di olio
    presente nel circuito frenante per l' invio della trap?"
    ::= {moto-threshold 6}

lowBrakePressureThreshold OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION "Definisce la pressione minima (espressa in bar) del
```

```

circuitto frenante per l' invio della trap"
::={moto-threshold 7}

usedPastillesThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Definisce lo spessore minimo (espresso in mm.) delle
pasticche dei freni per l' invio della trap"
::={moto-threshold 8}

highExhaustTempThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Definisce la temperatura massima (espressa in gradi
centigradi) dei gas di scarico per l' invio della trap"
::={moto-threshold 9}

highEngineTempThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Definisce la temperatura massima (espressa in gradi
centigradi) della testata del motore per l' invio della trap"
::={moto-threshold 10}

highWaterTempThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Definisce la temperatura massima (espressa in gradi
centigradi) del liquido di raffreddamento del motore per l' invio
della trap"
::={moto-threshold 11}

lowWaterLevelThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Definisce il numero di litri minimo di liquido di
raffreddamento presente nel motore per l' invio della trap"
::={moto-threshold 12}

highGearboxTempThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Definisce la temperatura massima (espressa in gradi
centigradi) della scatola del cambio per l' invio della trap"
::={moto-threshold 13}

highClutchTempThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current

```



```

DESCRIPTION "Definisce la temperatura massima (espressa in gradi
centigradi) della scatola della frizione per l' invio della trap"
::={moto-threshold 14}

highBrake1TempThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Definisce la temperatura massima (espressa in gradi
centigradi) del freno anteriore per l' invio della trap"
::={moto-threshold 15}

highBrake2TempThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Definisce la temperatura massima (espressa in gradi
centigradi) del freno posteriore per l' invio della trap"
::={moto-threshold 16}

-- TRAP DEFINITIONS

lowFuel NOTIFICATION-TYPE
OBJECTS {fuelLevel}
STATUS current
DESCRIPTION "Generata quando la quantita' di benzina scende
sotto lowFuelThreshold."
::={moto-trap 1}

highVoltage NOTIFICATION-TYPE
OBJECTS {voltage}
STATUS current
DESCRIPTION "Generata quando la quantita' di benzina sale
sopra highVoltageThreshold."
::={moto-trap 2}

lowVoltage NOTIFICATION-TYPE
OBJECTS {voltage}
STATUS current
DESCRIPTION "Generata quando la quantita' di benzina scende
sotto lowVoltageThreshold."
::={moto-trap 3}

lowOilLevel NOTIFICATION-TYPE
OBJECTS {oilLevel}
STATUS current
DESCRIPTION "Generata quando la quantita' di olio scende
sotto lowOilLevelThreshold."
::={moto-trap 4}

outOfRPM NOTIFICATION-TYPE
OBJECTS {engineRPM}
STATUS current
DESCRIPTION "Generata quando i giri del motore sono piu' alti del
livello outOfRPMThreshold."
::={moto-trap 5}

```

```

lowBrakeOil NOTIFICATION-TYPE
  OBJECTS {brakeOilLevel}
  STATUS current
  DESCRIPTION "Generata quando il livello dell' olio motore
scende sotto lowBrakeOilThreshold."
  ::= {moto-trap 6}

lowBrakePressure NOTIFICATION-TYPE
  OBJECTS {brakeOilPressure}
  STATUS current
  DESCRIPTION "Generata quando la pressione del circuito frenante
e' piu' basso di lowBrakePressureThreshold."
  ::= {moto-trap 7}

usedPastillesWarning NOTIFICATION-TYPE
  OBJECTS {brakePastilles}
  STATUS current
  DESCRIPTION "Generata quando lo spessore delle pastiglie e'
piu' basso dei usedPastillesThreshold."
  ::= {moto-trap 8}

highExhaustTemp NOTIFICATION-TYPE
  OBJECTS {exhaustTemp}
  STATUS current
  DESCRIPTION "Generata quando la temperatura dei gas di scarico
e' piu' alta di highExhaustTempThreshold."
  ::= {moto-trap 9}

highEngineTemp NOTIFICATION-TYPE
  OBJECTS {engineTemp}
  STATUS current
  DESCRIPTION "Generata quando la temperatura della testata e'
piu' alta di highEngineTempThreshold."
  ::= {moto-trap 10}

highWaterTemp NOTIFICATION-TYPE
  OBJECTS {waterTemp}
  STATUS current
  DESCRIPTION "Generata quando la temperatura del liquido di
raffreddamento e' piu' alta di highWaterTempThreshold."
  ::= {moto-trap 11}

lowWaterLevel NOTIFICATION-TYPE
  OBJECTS {waterLevel}
  STATUS current
  DESCRIPTION "Generata quando la quantita' del liquido di
raffreddamento e' piu' bassa di lowWaterLevelThreshold."
  ::= {moto-trap 12}

highGearBoxTemp NOTIFICATION-TYPE
  OBJECTS {gearBoxTemp}
  STATUS current
  DESCRIPTION "Generata quando la temperatura della scatola del
cambio e' piu' alta di highGearBoxTempThreshold."
  ::= {moto-trap 13}

```

```
highClutchTemp NOTIFICATION-TYPE
  OBJECTS {clutchTemp}
  STATUS current
  DESCRIPTION "Generata quando la temperatura della scatola della
frizione e' piu' alta di highClutchTempThreshold."
  ::= {moto-trap 14}

highBrake1Temp NOTIFICATION-TYPE
  OBJECTS {brake1Temp}
  STATUS current
  DESCRIPTION "Generata quando la temperatura del freno anteriore
e' piu' alta di highBrake1TempThreshold."
  ::= {moto-trap 15}

highBrake2Temp NOTIFICATION-TYPE
  OBJECTS {brake2Temp}
  STATUS current
  DESCRIPTION "Generata quando la temperatura del freno
posteriore e' piu' alta di highBrake2TempThreshold."
  ::= {moto-trap 16}

END
```

## **Conclusioni**

Questo progetto ha lo scopo di fornire una base concettuale per un possibile utilizzo del protocollo SNMPv2 anche per applicazioni critiche come quelle di una gara motociclistica. Sono stati trascurati tutti gli aspetti di interfacciamento dei sensori con la centralina e l'implementazione dell'agent su tale piattaforma (anche per mancanza di tempo e di materiali). In ogni caso il mib prodotto e' stato testato ed e' possibile una sua implementazione reale.

## **Sviluppi Futuri**

Purtroppo, per mancanza di tempo non e' stato possibile approfondire tutte le applicazioni della centralina. Si sarebbe potuta introdurre la possibilita' dei meccanici di regolare il motore sempre tramite SNMP e un controllo piu' approfondito per l'interfaccia con il modulo 802.11.

## **Riferimenti**

J.Schonwalder, L.Deri "Sistemi di elaborazione dell'informazione".  
RFC 1155.

[www.steroidmicro.com](http://www.steroidmicro.com) – Produttori della centralina

[www.ietf.org](http://www.ietf.org) – Riferimenti per il MIB

[www.ducati.it](http://www.ducati.it) – Immagini

Il MIB e' stato testato sul sito: [www.simpleweb.org/ietf/mibs/validate](http://www.simpleweb.org/ietf/mibs/validate)