

Do-Dots

Guida alla scrittura

Ultimo aggiornamento - 5 agosto 2010

rev1 - Stesura iniziale - 14/05/2010



do-Dots – Guida alla scrittura dei Dot

Introduzione

Do-Dots è un framework che facilita la comunicazione e l'integrazione di sistemi di automazione. E' basato su un protocollo comune a livello applicativo che permette la comunicazione in una rete composta da tanti Dot e un Broker.

I Dot sono piccoli componenti che possono implementare interfacce, logiche e tradurre protocolli esterni.

Il Broker è un componente centrale che si assicura che le informazioni siano condivise velocemente tra un Dot e un altro.

Ogni Dot è visibile non come una tabella di registri o indirizzi difficile da interpretare, ma come un 'oggetto' leggibile e auto descrittivo che espone variabili, metodi ed eventi.

Il risultato è un kit di strumenti che facilitano la gestione e l'estensibilità dei vostri sistemi di automazione, permettendovi di scrivere il minimo indispensabile e avvantaggiandovi della compatibilità con tanti Dot disponibili nella nostra libreria.



Perché scrivere un Dot

Scrivere un Dot è il passo iniziale e fondamentale per collegare il proprio sistema al framework do-Dots.

Per scrivere un Dot si possono seguire due strade: *implementare direttamente il protocollo di do-Dots* e inviare e ricevere pacchetti su un socket, oppure *appoggiarsi a progetti che nascondono la comunicazione vera e propria attraverso classi da derivare*.

Scaddle trainer segue la seconda strada.

Quale tecnologia è supportata ?

A questo stato dello sviluppo le interfacce grafiche, i Dot e il Broker sono scritti in per il Microsoft .NET Framework versione 2.0 o successiva.

Non essendo il .NET Framework multipiattaforma, stiamo lavorando per portare la struttura di do-Dots anche in altri sistemi operativi.

Gran parte del codice gira perfettamente su Mono, e stiamo pianificando un passaggio a Qt per la maggior parte dei componenti.

Dove installare un Dot ?

L'architettura di do-Dots permette di scegliere su quale hardware installare Broker, interfacce utente, e Dot.

Se un dispositivo è collegabile direttamente in Ethernet ed implementa uno stack TCP, è un buon candidato per contenere un Dot.

Dispositivi con maggiore potenza di calcolo possono eseguire Dot, Broker e interfaccia utente. Un mini-PC dotato di touchscreen può gestire una rete di dispositivi e contemporaneamente fornire un'interfaccia utente per il controllo.



Scrivere un Dot con Scaddle trainer

Introduzione

Scaddle trainer è uno strumento che si pone a metà strada tra la demo e lo strumento di training.

Permette infatti di avere una dimostrazione semplificata di una rete do-Dots ma anche di poter interagire modificando direttamente uno dei Dot della rete.

Scaddle trainer è volutamente limitato nelle funzioni e nella simulazione, per evitare che troppe informazioni o una complessità eccessiva possano confondere le idee.

Nella versione attuale, ad esempio, sarà possibile solamente leggere e scrivere il valore delle variabili esposte dai vari Dot.



Modificare il Dot vuoto

Gli esempi seguenti prevederanno che si apra la soluzione Scaddle trainer tramite una versione di Visual Studio e si modifichi il file sorgente **MyDot.cs**.

Ad ogni compilazione ed esecuzione, verrà lanciato Scaddle trainer con le modifiche apportate al Dot.

Tramite questa guida, sarà possibile seguire un percorso che porterà al test di varie funzioni del sistema.

Esempio - Definire e modificare le variabili

E' comune per un Dot esporre il contenuto di variabili nella rete.

Le variabili possono essere aggiunte, rimosse e modificate in qualunque momento all'interno del codice del Dot.

Questo esempio prevede che si inizializzi il Dot con due variabili di esempio.

Per farlo, occorre sovrascrivere il metodo `OnInstall()` nel file `myDot.cs`.

All'interno del corpo del metodo, aggiungere alcune variabili al Dot con le seguenti righe:

```
public override void OnInstall()
{
    base.OnInstall();

    this.Variables.Add("Tipo", "training");
    this.Variables.Add("Counter", 0);
    this.Variables.Add("NumLightsOn", 0);
}
```

Per verificare l'aggiunta delle variabili aprire dal menu principale di Scaddle trainer la pagina Dots, selezionare MyDot e verificare nella lista delle variabili se il MyDot contiene le variabili aggiunte tramite codice.



Esempio – Leggere le variabili di un altro Dot

Leggere le variabili di un altro Dot è un'operazione quasi necessaria per un Dot che deve gestire una logica.

Lo stato delle variabili viene ritornato direttamente dal Broker, che lo memorizza e lo mantiene sempre aggiornato.

Inserire le seguenti righe nel file `myDot.cs`:

```
public override void OnTick()
{
    base.OnTick();

    int numLightsON = 0;

    if (ReadVariable("Office/Light1") == "1")
    {
        numLightsON++;
    }

    if (ReadVariable("Office/Light2") == "1")
    {
        numLightsON++;
    }

    WriteVariable("NumLightsOn", numLightsON);
}
```

L'esempio mostrato fa in modo che, regolarmente, il Dot richieda lo stato delle variabili 'Light1' e 'Light2' del Dot 'Office'.

Il Dot quindi conta quante di queste variabili valgono 1 e mette il risultato nella propria variabile 'NumLightsOn'.

Il risultato è che il Dot modificato è in grado di contare quante luci sono accese nel Dot 'Office'.



Esempio – Modificare le variabili di un altro Dot

L'implementazione in .NET del Dot permette di modificare le variabili locali o globali in maniera trasparente.

Prendendo ad esempio questo codice:

```
// Increments a counter:  
if (++this.MyCounter > 1000)  
{  
    this.MyCounter = 0;  
}  
  
WriteVariable("Counter", this.MyCounter);
```

Si può notare che la chiamata a **WriteVariable** scrive un nuovo valore nella variabile 'Counter' interna al Dot.

L'effetto è che si può notare un contatore che incrementa costantemente fino a 1000 per poi azzerarsi.

Aggiungendo la riga evidenziata di seguito, si può vedere come scrivere un valore *in un altro Dot*:

```
// Increments a counter:  
if (++this.MyCounter > 1000)  
{  
    this.MyCounter = 0;  
  
    WriteVariable("Office/Light1", 1);  
}
```

La nuova riga scrive il valore '1' nella variabile 'Light1' del Dot 'Office', ed equivale ad accendere una luce nel momento in cui il contatore del 'MyDot' raggiunge il valore 1000.