GorAcquire

Indice

Sommario

[Scopo e campo di applicazione 2](#_Toc413408239)

[Documenti di riferimento 2](#_Toc413408240)

[Strumentazione, dispositivi e apparecchiature utilizzate 2](#_Toc413408241)

[Condizioni di utilizzo 2](#_Toc413408242)

[Operazioni preliminari 2](#_Toc413408243)

[Operazioni da svolgere (procedimento operativo) 2](#_Toc413408244)

[Registrazioni 2](#_Toc413408245)

[Operazioni conclusive 3](#_Toc413408246)

# Tabella delle revisioni

|  |  |  |  |
| --- | --- | --- | --- |
| n.rev. | Data di emissione | Preparata da nome/firma | Approvata da Nome/firma |
| 1 | 27-02-2015 | Francesco Lippolis | Francesco Lippolis |
| 2 | 10-04-2015 | Francesco Lippolis | Francesco Lippolis |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Scopo e campo di applicazione

Lo scopo è di creare una classe GORACQUIRE il quale serve per memorizzare in un database tutti i dati che la nostra centralina dovrà leggere tramite i sensori.

# Documenti di riferimento

Norme: nessuna

Documenti interni: nessuna

Altri: nessuna

# Strumentazione, dispositivi e apparecchiature utilizzate

* Pc contenente DataBase
* RaspBerry Pi
* Sensori di luce, umidità, temperatura
* Pc per stesura codice C# con preIstallato Visual Studio 2013

# Condizioni di utilizzo

Per poter implementare la classe GORACQUIRE è indispensabile avere uno schema del database con un diagramma ER.

# Operazioni preliminari

Come prima fase si è dovuto decidere e implementare un database, ed stato opportuno decidere come doveva memorizzare i vari dati sul database. Una volta deciso si è steso al diagramma ER per poter creare il database fisico. Una volta costruito il database su un pc si è iniziata la creazione della classe in C# per la scrittura dei dati su database.

# Operazioni da svolgere (procedimento operativo)

Per poter utilizzare la classe GORACQUIRE si deve essere in un possesso di un database già creato su una macchina.

Prendere la stringa di connessione del database e creare l’oggetto GORACQUIRE.

Una volta creato tramite il metodo SaveMeasurement(Measurement m)e possibile scrivere nel database, e passandoli un oggetto Measurement m, dove conterrà tutti i dati letti dalla centralina.

# Codice Classe GorDbWriter

**public class GorDbWriter**

**{**

**OleDbConnection connection;**

**OleDbCommand command;**

**int startRead = 0;**

**int endRead = 100;**

**string connectionString = "METTERE QUA LA CONNECTION STRING";**

**List<Sensor> Sensori;**

**public GorDbWriter()**

**{**

**try**

**{**

**//creazione oggetto e passaggio stringa di connessione**

**connection = new OleDbConnection(connectionString);**

**}**

**catch**

**{**

**throw new Exception();**

**}**

**}**

**/// <summary>**

**/// Salva una misurazione unica (su un solo sensore)**

**/// </summary>**

**/// <param name="IdDatabase">Primary key nel database Garden of Things</param>**

**/// <param name="m">Misura effettuata sul sensore che si deve salvare</param>**

**/// <returns></returns>**

**public bool SaveMeasurement(string IdDatabase, Measurement m)**

**{**

**string id="";**

**try**

**{**

**//Creazione stringa data + tempo ( Anno , mese, giorno , ora, minuti, secondi )**

**string datatimenow = m.SampleTime.ToString("yyyy-MM-dd HH:mm:ss");**

**//creazione query SQL**

**string query = "INSERT INTO Rilevazione (Tempo,Valore,PuntoMisura) VALUES('" + datatimenow + "'," + m.Value.ToString() + "," + id + "')";**

**command = new OleDbCommand(query, connection);**

**return true;**

**}**

**catch**

**{**

**return false;**

**}**

**}**

**public bool SaveAll(List<Sensor> Sensori)**

**{**

**foreach (Sensor s in Sensori)**

**{**

**string IdSensore = s.CodiceGardenOfThings;**

**Measurement m = s.LastMeasurement;**

**SaveMeasurement(IdSensore, m);**

**}**

**return true;**

**}**

**/// <summary>**

**/// Prende da file i dati quando la rete non funziona**

**/// </summary>**

**/// <param name="percorso">Percorso e nome del file del log dei dati</param>**

**public void PrendiDaFile(string percorso)**

**{**

**//TODO: il programma deve fare:**

**// XXXX**

**try**

**{**

**// vedere se questo codice si può riutilizzare**

**using (StreamReader sr = new StreamReader(percorso))**

**{**

**char[] buffer = new char[endRead];**

**sr.Read(buffer,startRead,endRead);**

**string lettura="";**

**for(int i= 0; i < buffer.Length -1; i++)**

**{**

**lettura = lettura + buffer[i];**

**}**

**string[] slitlettura = lettura.Split(' ');**

**Measurement sensore = new Measurement();**

**//DA FINIRE**

**}**

**}**

**catch**

**{**

**throw new Exception();**

**}**

**}**

**}**

# Registrazioni

E’ stato eseguito un push su git. <https://github.com/gamondue/GOR5F>

# Operazioni conclusive

Per verificare se la classe funziona si sono fatte delle prove di lettura. La scrittura non è stato possibile verificarla, perché non vi erano gli accessi a un database creato.