مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (30) العدد (5) 2008
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (30) No. (5) 2008

# Using Supervisory Control And Data Acquisition (SCADA) System in the Management of a Diesel Generator

**Dr. Ebraheem Alshami** *
**Dr. Hassan Albustani** **
**Mohammed Melhem*****

**(Received 28 / 2 / 2008. Accepted 2 / 4 / 2008)**

## □ ABSTRACT □

Supervisory Control And Data Acquisition (SCADA) systems have been developed to reduce costs by optimizing plant operations. They allow wide monitoring and remote control from a central location. SCADA system is widely used in critical infrastructures such as electrical grid, natural gas, petroleum fields, wastewater industries, and other public infrastructure systems. SCADA system provides monitoring, data analysis, and control of equipment. In this paper, we present the technologies for data collection, data storage, and reporting in SCADA System. We also show and how these systems manage time by giving an overview of a SCADA system architecture and the functions it performs. Then, we discuss a practical implementation of a SCADA system taken from AL-ZARA Thermal Power Station.

**Keywords**: SCADA, HMI, RTU, PLC, IED, Citect, Data Acquisition, Diesel Generator Management.

* **Associate Professor, Department of Computers and Control, Faculty of Mechanical and Electrical Engineering, Al-Baath University, Homs, Syria.**
****Assistant Professor, Department of Communication, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**
*****Postgraduate Student, , Department of Computers and Control, Faculty of Mechanical and Electrical Engineering, Tishreen University, Lattakia, Syria.**

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (30) العدد (5) 2008
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (30) No. (5) 2008

# استخدام نظام المراقبة والتحكم وتحصيل المعطيات (SCADA) في إدارة مولد ديزل

**الدكتور إبراهيم الشامي***
**الدكتور حسن البستاني****
**محمد ملحم*****

**(تاريخ الإيداع 28 / 2 / 2008. قُبِل للنشر في 2008/4/2)**

## □ الملخّص □

وجدت نظم المراقبة والتحكم وتحصيل المعطيات (SCADA) من أجل خفض الكلفة عن طريق أمثلة العمليات في المنشأة الصناعية، بالإضافة إلى أن هذه النظم تعطي إمكانية المراقبة الواسعة والتحكم عن بعد من موقع مراقبة مركزي. كذلك تستخدم نظم SCADA بشكل واسع في إدارة البنى التحتية العملية كشبكات الكهرباء، والغاز الطبيعي، و حقول النفط و المياه الصناعية، وغيرها من البنى الأساسية، وكذلك تؤمن جمع وتحليل المعطيات والتحكم بالمعدات.

سنقدم من خلال هذه الدراسة شرحاً لتكنولوجيا تحصيل المعطيات وطريقة تخزينها و تحليل المعلومات و تشكيل التقارير في نظم SCADA، كما سنوضح كيف تقوم هذه النظم بإدارة الوقت، من خلال شرح البنية الأساسية لنظم SCADA، والوظائف التي تقوم بها، ومن ثم سوف نطرح تطبيقاً عملياً لنظام SCADA مأخوذاً من محطة التوليد الكهربائية في الزارة.

**الكلمات المفتاحية:** SCADA, HMI , RTU , PLC , IED , Citect , Data Acquisition , Diesel Generator Management.

*أستاذ مساعد – قسم التحكم والحواسيب–كلية الهندسة الميكانيكية و الكهربائية–جامعة البعث– حمص–سورية.
**مدرس – قسم الاتصالات– كلية الهندسة الميكانيكية و الكهربائية–جامعة تشرين– اللاذقية–سورية.
***طالب دراسات عليا (ماجستير) – قسم الحاسبات والتحكم– كلية الهندسة الميكانيكية و الكهربائية–جامعة تشرين– اللاذقية–سورية.

## Introduction:

Computer-based control systems are used in many infrastructures and industrial processes (e.g., electrical grid, natural gas, water…) to monitor and control sensitive processes and physical functions. These systems enable an operator to remotely view real-time measurements, such as level of water in a tank, and remotely initiate the operation of network elements such as pumps and valves. SCADA systems can be set up to sound alarms at the central host computer when a fault in a water supply system is identified[1].

SCADA systems collect sensor measurements, operational data from the field process and display this information. Also they relay control commands to local or remote equipment. They are used to keep a historical record of the temporal behavior of various variables in a system such as tank and reservoir levels. By SCADA system, we decrease local monitoring and depend only on sensor signals, which are very useful in dangerous and large geographical processes, such as nuclear or petrol fields, gas pipelines, and traffic. As shown in figure(1) (10 years site & sensor projection in USA) [2].

On the other hand, Heavy Industrial Companies use Distributed Control Systems (DCS) in single processing or generating plants over a small geographical area. But SCADA systems are used for geographically large and dispersed operations. For example, a utility company may use a DCS to generate the power and a SCADA system to distribute it. As presented here in Syria, AREVA, a heavy industrial Company, uses SCADA system to manage the local electrical grid [3], it links electric dispenser centers (Substations) and generation stations, to the Automatic Dispatch Center(ADC) in Damascus, by using Telemetry Dialing Systems, Fiber Optic Cables**,** RTU (Remote Telemetry Unit) as I/O devices and e-terra/SCADA software .

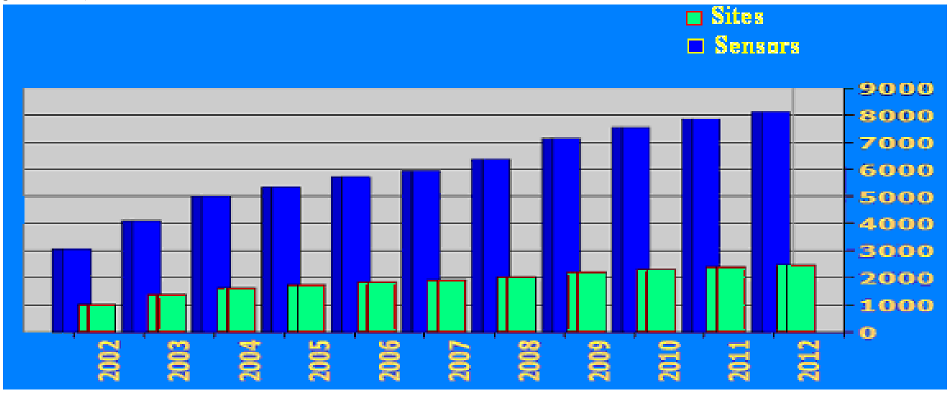



**Figure (1): 10 years site & sensor projection**

As mentioned above, SCADA system reduces costs; this point appears clearly in figure(2) [4] in that it shows the cost difference between SCADA and DCS system by the count of physical I/O points. It shows that the costs of DCS approximate the costs of the SCADA system when I/O points count is greater than 1200 physical I/O. Our example that will be discussed later is the monitoring and control of a diesel generator[5].

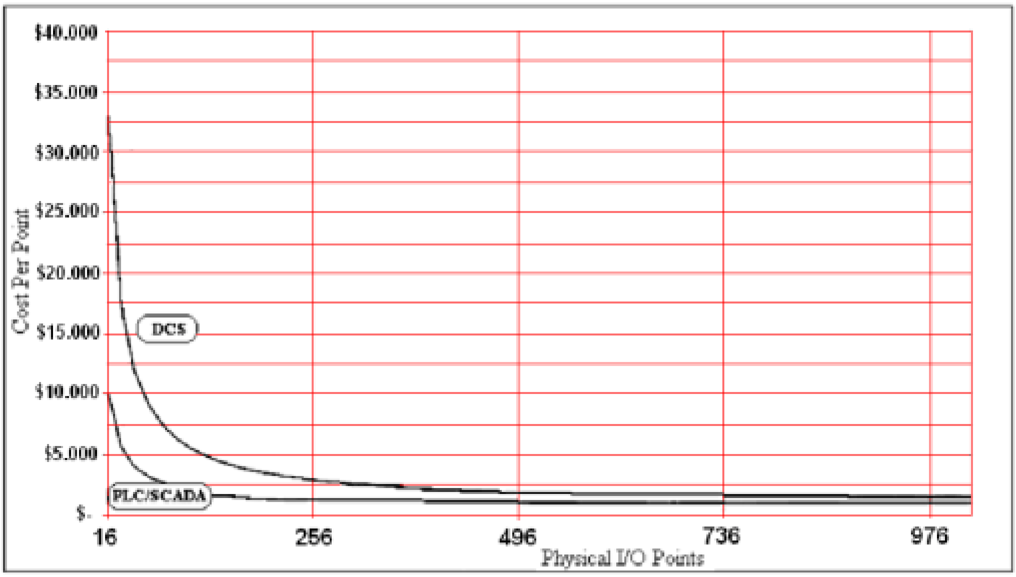



**Figure (2) :Costs Comparison Of Control Systems Overview**

This system has about 22 physical I/O points. The cost of this control system is approximately 10.000$ if we use a DCS system and about 2500$ for SCADA system. the cost of the SCADA is 75% less. The SCADA system will be built depending on Citect/SCADA as SCADA software and on Siemens Programmable Logic Controller (PLC), "Simatic5 (S5-100U)" as Input/Output Device [6],[7].

## The Purpose and Importance of Research

The purpose of research is to show the facilities that SCADA system provides in controlling and monitoring, specially in the industrial process where reliability and real time monitoring are needed.

## 1. SCADA System Analysis and Functionality

SCADA is a flexible and scalable system. It can start out as simple as we like, and grow as large as required. In this section we discuss the architecture from two points: hardware and software.

### 1.1. Hardware Architecture

SCADA system consists of three parts [8], [9] as shown in figure(3).

**Input/Output Devices**: These devices are connected to the field like Programmable Logic Controller(PLC), Remote Telemetry Unit (RTU), Intelligent Electronic Device( IED) and other controllers. These controllers are industrial gradable programmable devices that perform the manual and automatic control functions associated with one or more plant processes and solve important calculations such as flow totalizations and runtime totalizations, and reading sensor measurements (flow, pressure, temperature, level, pollution…etc). Field equipments is typically hardwired to the PLC(I/O device) via I/O modules.

**Input/Output Server**: Central host computer that consists of I/O Server, historical data server, Human Machine Interface (HMI) clients, historical data client [8].

•HMI client and HMI I/O server: The HMI client is a graphical user interface that allows operators to monitor and control plant equipment by viewing graphics, clicking on buttons, and entering process setpoints information. The HMI client allows the user to view and manipulate the information that is available in the PLC(I/O device). The HMI I/O server performs several important functions.

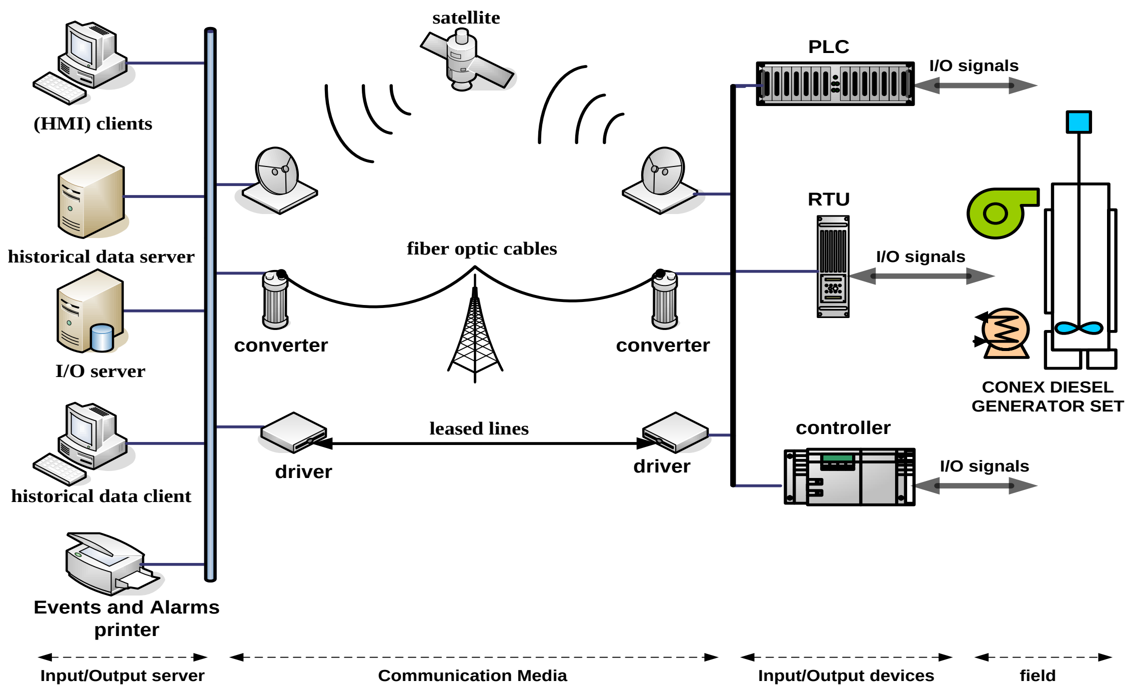
1- It polls data from the (I/O) Devices and updates the HMI client graphics with newly acquired information .

2- It transfers all updated operator commands and setpoints from the HMI graphics to the (I/O) Devices immediately after the change is made.

3- It provides updates of the historical data server, taking into account the problem of failing of the HMI I/O server. By this update, we save the data at the time of operation; therefore, redundant I/O servers are typically implemented. If a primary I/O server fails, the standby server is available to provide data to the HMI graphics and historical data server.

•Historical data server: The historical data server collects information from the I/O servers and stores it for future use. The architecture of the historical data server depends on the required level of redundancy.

•Historical data client: The historical data client is a machine that has the facility to access historical data, usually in the form of graphical trends or reports.




**Figure(3) : SCADA system**

**Communication media:** at last count, there are well over 200 SCADA protocols[9] in use around the world for communication between a central computer (I/O server) PLC, RTU (I/O device) and flow computers. The most modern SCADA system uses one or more of these virtual standard protocols:

- ASCII (American Standard Code for Information Interchange).
- CAP (Compressed ASCII Protocol).
- Modbus, an ubiquitous point to point PLC protocol.

- ModbusX, an expansion of the Modbus protocol.
- IEEE 32 bit single format floating point.
- DNP3, IEC 60870.5 [10].

Other industrial protocols are carried over several kinds of communication media. Such as wire communication, wireless communication and optical communication. We can review these three kinds in five subsystems which are listed in order of popularity [9]:

1- communication over cables or leased lines.
2- communication over fiber optic cables.
3- Telemetry dialing systems.
4- communicating over the Internet.
5- communicating over satellites.

## 1.2. Software Architecture

SCADA software can be divided into two types: proprietary or open. Companies develop proprietary software to communicate with their hardware. Open software is able to mix different manufacturers' equipment in the same system. Citect and WonderWare SCADA are just two of the open software packages available on the market for SCADA systems[10]. SCADA handle three functions:1-**S**upervisory 2-**C**ontrol 3-(**A**nd) **D**ata **A**cquisition. These three functions are translated by the SCADA central station computer into numbers and Boolean conditions in SCADA Real Time Data Base (RTDB) [9]; this data base is generally in the shape of long list of variable tags. In modern object oriented software, like National Instrument's Lookout, Citect/HMI SCADA and Wonder Ware (RTDB) located in one or more servers .Servers are responsible for data acquisition and handling (polling controllers, alarm checking, calculations, logging and archiving). As shown in figure(4) the data flow between client and I/O server[11],[12]. showing the two processes (monitoring and control).

### 1.2.1.Monitoring

As shown in figure (4), if any changes occur, the driver writes a new value to the RTDB. The Control Server manages real-time database access and triggers off coupled server every instance the Process Value (PV) is changed in the RTDB. The Calculation server converts ADC codes (raw values) into real physical (operational) values and writes them to RTDB. The Visualization Server gets information about changes of operational values and appends these data changes to message queue driven by Message Queue server. HMI cyclically checks for the messages in queue, reads them out and represents them by graphical image.

### 1.2.2.Control

The Operator controls the process via HMI; it presses buttons, enters new values, etc. HMI writes new values to RTDB. The Calculation Server converts the operational values (volt, Amper, etc.) into DAC code and writes them to RTDB (raw values part). Next, the control Server sends message to coupled driver.

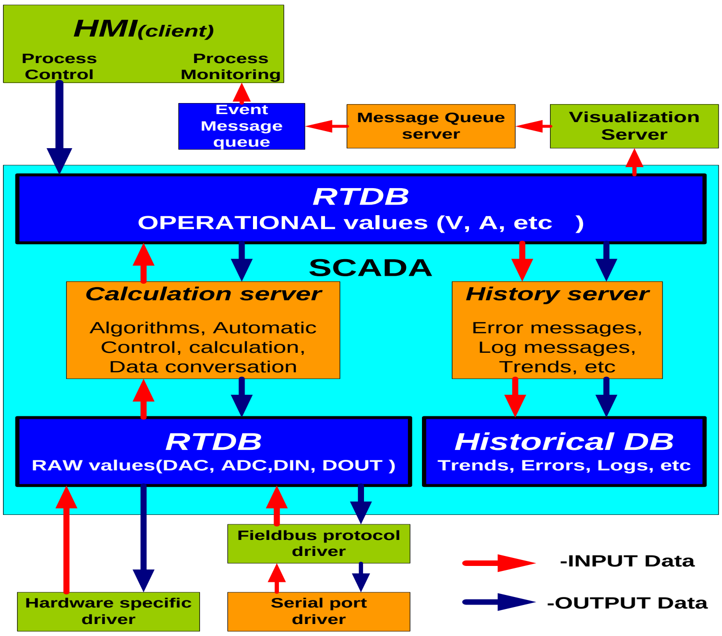



**figure (4): data flow**

## 1.3. SCADA Functions

There are many functions that can be handled by SCADA system. We present the most popular ones:

1. *Input/Output*

This is the ability to accept and transmit data between the supervisory computer(s) and I/O devices (PLC, RTU…) .

2. *HMI*

The products support multiple screens which can contain combinations of synoptic diagrams and text. They also support the concept of a "generic" graphical object with links to process variables. These objects can be “dragged and dropped” from a library and included into a synoptic diagram.

3. *Alarm Handling*

Alarm handling is based on limit and status checking and performed in data servers. The alarms are logically handled centrally; the information only exists in one place and all users see the same status; multiple alarm priority levels are supported. It is generally possible to group alarms and to handle them as an entity. Furthermore, it is possible to suppress alarms either individually or as a complete group.

4. *Event Handling*

An Event is handled the same as an alarm, but is based on happening checking (pump start, pump stop, valve open ,..etc). And there is no need to suppress events.

5. *Trending*

The SCADA software provides trending facilities and capabilities as follows:

- The parameters to be trended in a specific chart can be predefined or defined on-line.
- Real-time and historical trending are possible.

6. *SPC*

Statistical Process Control (SPC) is a method of analyzing and controlling the quality of material, manufacturing, products, services, etc. Also, we can use SPC charts(trends).

7. *Logging/Archiving*

Logging and archiving are often used to describe the same facilities. However, logging can be thought of as medium-term storage of data on disk, whereas archiving is long-term storage of data either on disk or on another permanent storage medium.

8. *Report Generation*

Report can be built by using SQL type queries in the archive, in real time database (RTDB); it is also sometimes possible to embed EXCEL charts in the report.[11], [6] .

## 1.4. Polling Intervals and System Performance

Getting information quickly and reliably to and from the plant floor is the primary role of any SCADA system. Therefore, many SCADA systems employ some form of data acquisition scheduling to conserve bandwidth. Consequently, data are not often collected continuously from all I/O devices in the field; they display analog data as some form of average value rather than continuous instantaneous values. There are two major ways of obtaining data from the field [2]:

**1-**The central host computer (I/O server) "polls" the field devices (I/O devices):

I/O server polls the I/O devices to report the latest analog data values, alarm and equipment status (pumps on/off, valve open/close…). If no change of state has occurred polled I/O devices can report by exception, in which case they respond with "nothing to report". If a change of state has occurred, the I/O devices report the appropriate information. This approach reduces the bandwidth requirement of the system.

**2-**The field devices send "unsolicited" data to central host (I/O server):

I/O devices generate all reporting messages as required in a "random" fashion. Typically, such messages report a change of state or a fault, or simply pass data to the I/O server. By these strategies, SCADA's communication is demand-based, reading only those points which are requested by clients. On the other hand, if all requested data are grouped together, then fewer requests are required, and response is faster.

As following CitectSCADA uses a blocking constant to calculate whether it is quicker to read them separately or in the same 'block'. By compiling a list of the registers that must be read in one scan, CitectSCADA automatically calculates the most efficient way of data reading figure(5) [6] .

**Example**: CitectSCADA requires registers 1012 and 1020. The I/O device has a read overhead of 60ms- which is independent of the number of registers read.

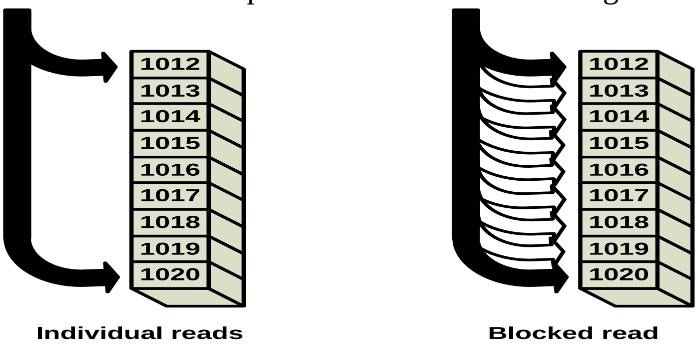



**Figure (5): registers reading**

Individual reads:
Protocol request = 8 bytes Transmit time = 7ms @ 9600 baud
Protocol response = 7 bytes Transmit time = 6ms @ 9600 baud
Total response time = (7 + 60 + 6)x2 = 146ms
Blocked read:
Protocol request = 8 bytes Transmit time = 7ms @ 9600 baud
Protocol response = 23 bytes Transmit time = 19ms @ 9600 baud
Total response time = 7 + 60 + 19 = 86ms
By this method (Blocked read) SCADA reduces reading time approximately to half.

### 1.5. Simple SCADA System

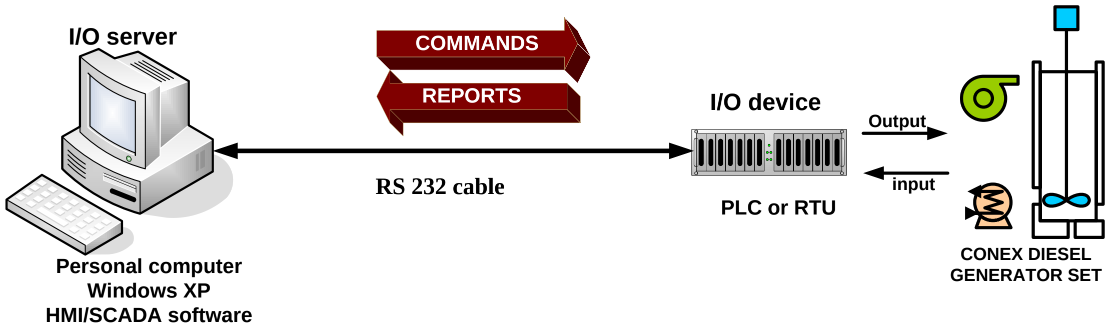
In the previous sections, we have defined the SCADA in general and complicateed form (large system over 100.000 tags)[13]. Now we'll show the simple form of SCADA system. which only consists of three simple components, as shown in figure(6).

1-Personal computer runs a windows XP as an operation system and HMI/SCADA software. In this case, we have approximately about 1000 tags; that is, one operator can monitor the industrial process. Thus, there is no need to build a network as an operation station; only one computer with windows XP and SCADA software as I/O server.

2-Communication media is a serial communication cable which is leased lines like RS232 cable for short distance, but for long distance, it is better to use a fiber optic cable because it is more reliable than the other.

3-PLC (Programmable Logic Controller) as I/O device .

As noted above, we use PLC because it performs the best: control industrial processes. Notice that we can use RTU because it performs the best communication measurement and commands to and from a remote site [9]. This form of SCADA system is very convenient to use in our industrial implementation (monitoring and control of a diesel generator).




**Figure(6): SCADA system in simple form**

## 2. A SCADA System for Managing a Diesel Generator

From figure (3), the field that will deal with it is a CONEX diesel generator set, which is an emergency generator in AL-ZARA thermal power station.

### 2.1 Diesel Generator Specifications

Type (2000KB16V45), dimensions (2200x1204x1846)mm, weight (2740 Kg), number of cylinders(16), speed (1500 rpm), operation method (four-stroke cycle, single-acting) and combustion method (direct injection), generator (1500 rpm (50HZ) 6 poles generators ), capacity 936 KVA, continuous power (795 KW at 1500 rpm ), …etc [5] .

## 2.2. Diesel Generator Controlling

The generator has a Detroit Diesel Electronic Control System (DDEC III), which is an advanced technological electronic fuel injection and control system. DDEC III system provides the capability to protect the engine from serious damage resulting from conditions such as high engine temperature or low oil pressure.

The major subsystems of DDEC include:

1.Electronic control module (ECM).

2.Electronic unit injection (EUI).

3.Engine sensors.

ECM receives electronic inputs from sensors on the engine; it also contains an Electrical Erasable Programmable Read Only Memory (EEPROM). The EEPROM contains programs that control the basic engine functions, such as rated speed and power, timing of fuel injection, diagnostics, cold start logic, engine governing and engine protection, etc. Engines with more than eight cylinders operate with multiple ECMs. One ECM is called master, while the others are referred to as receivers, in our case, two ECMs. The master ECM is the controller of the engine; it receives input from the various sensors and sends information to the receiver ECM. Fuel is delivered to the cylinders by the Electronic Unit Injection (EUI), which is cam-driven to provide the mechanical input for pressurization of the fuel. The ECM controls solenoid operate valves in the EUIs to provide precise fuel delivery.

The sensors that can be provided with the DDEC engine are:

1-Fuel pressure sensor . 2-Oil pressure sensor . 3- Fuel temperature sensor .

4-Coolant temperature sensor . 5- Oil temperature sensor .

As shown in the figure(7), the general view of connecting engine sensor wiring harnesses to the receiver ECM and master ECM.

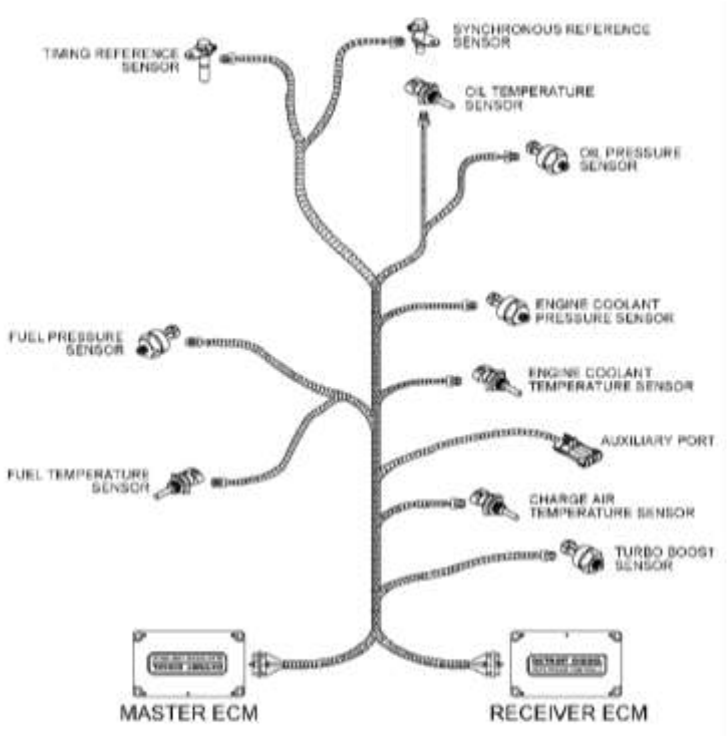



**Figure (7): general view of connecting engine sensors wiring harness to the receiver ECM an master ECM**

DDEC III has twelve digital input ports(451, 542, 528, 523, 541, 544, 543, 524, 531, 583, 545, 979) located on the Vehicle Interface Harness. These digital inputs can be configured for various functions. Some of the digital input functions are listed below:

◙-Engine protection. ◙- Air compressor controls.

◙-Pressure Governor System (PGS)(uses five inputs), etc.

Also DDEC III has six digital output ports (988,555,499,563,564,565). These digital outputs can be configured for various functions. Some of the digital output functions listed below:

◙-Coolant level low light. ◙- Vehicle power shutdown. ◙-Low oil pressure light, etc.

## 2.3. SCADA System Designing

In the previous section, we have illustrated how DDEC III collects the sensor signals, protects and controls the engine. To design a simple SCADA system, we have to know three points:

1. the kind of I/O device, in our case it is PLC (Programmable Logic Controller). PLC: siemens simatic S5-100U [7],[14].
2. communication media is RS232 cable using protocol AS511 [15].
3. SCADA software is CitectHMI/SCADA; it is an Open software [16].

Also we need to have information about several points on the diesel generator. Some measurement points are analog and the others digital. I/O modules (digital and analog) transfer these measurement to main control program in the PLC (S5-100U).

There are two kinds of signals:

1. One already exists; that is, it come from digital output of DDEC III on the engine.
2. The other needs to be added like (speed sensor, fuel level sensor, coolant level sensor, generator temperature, circuit breaker (open/close), etc).

At this point, we have two levels of control:

1. DDEC III protect and control the engine individually.
2. PLC(S5 100U) control the diesel generator (engine + generator ) individually.

The third level that will be added is the SCADA level (communication media + SCADA software). By three control levels, we have a SCADA system as shown in figure(8) .

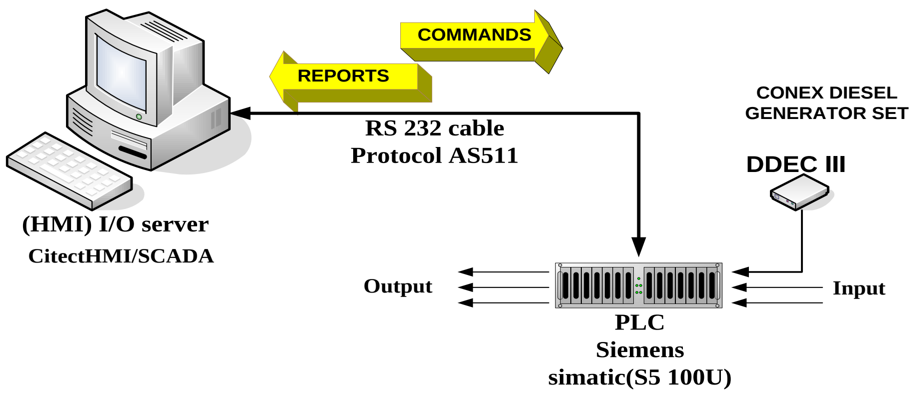



**Figure (8): SCADA system using CitectHMI/SCADA as software, RS232 (protocol AS 511) as communication media and PLC(S5 100U) as I/O device.**

In the memory (RAM) of PLC(S5-100U), two area process image area and interface area. All inputs and outputs are stored in the process image area, which is Process Image Input (PII) and Process Image Output (PIO) [7].

Data blocks (DBs) are the main stored units in the interface area and flags (temporary storage). PLC (S5-100U) have maximum 256 DBs, ordered from DB0 to DB255. We use them to store data (commands, reports ).

Main control program ”diesel generator control program” in PLC collects these data from PII and the DB(SCADA commands) and updates PIO, DB(SCADA reports). Protocol AS511 provides citectHMI/SCADA with required data from the interface area (DB).

## 2.4. Building of Operator Interface System(OIS)

**Step1**:

To build (OIS) [6] using CitectHMI/SCADA, which is the application that will be operated through the keyboard and mouse located at the CitectHMI/SCADA server, we need to arrange all process data that will be needed in control and monitor, as shown in table(1), the process data table.

Column1: variable from field (diesel generator).

Column2: PLC variables that the OIS will read and write .process variables in PLC registers(DBs) to control and monitor the generation area, and their address in PLC and read/write(report/command) status.

Column3: citectHMI/SCADA variable Tags.

Column4: the ranges for integer variables.

Column5: show how each variable will be displayed.

**Step2**:

In this step, we assess the variables address in PLC and the communication

protocol. As we have mentioned above, our I/O device is PLC(S5-100U) and SCADA software is citecHMI/SCADA which supports serial communication protocol (siemens AS511 protocol). AS511 is an industrial protocol usually used with siemens’s PLC ( simatic S5). This protocol is made up of 10 functions. The main functions are[15]:

1-FUNCTION BLOCK INFORMATION (B_INFO)

Input: block number. Output: initial absolute address of the block in PLC memory

2- DATABLOCK READ FUNCTION (DB_READ)

Input: initial absolute address in PLC memory, final absolute address in PLC memory
Output: contents of datawords.

3- DATABLOCK WRITE FUNCTION (DB_WRITE)

Input: initial address in AG(PLC), contents of datawords. Output: none.

4- SYS_PAR FUNCTION

Input: none. Output: address of I/O, flags, counter and timers.

As we have illustrateed in the beginning of this section, data blocks are the main units to store the data needed in citecHMI/SCADA server, and followed by the addressing method.

Db: w BCD / INT / LONG / LONGBCD / REAL / STRING.

Db: w.n DIGITAL.

Where

b is the data block number (as configured in the PLC).

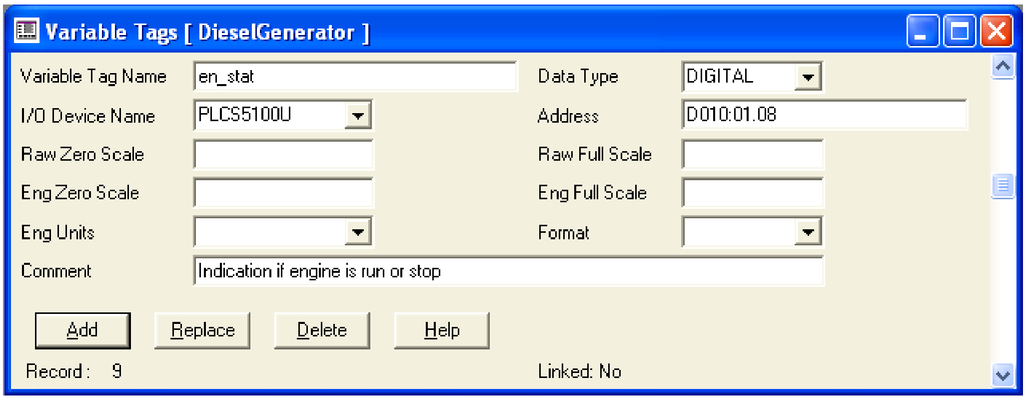
w is the word number…

n is the bit number 00 to 15.

And the address associated with the variable tags in citectHMI/SCADA in figure(9) for digital variable.

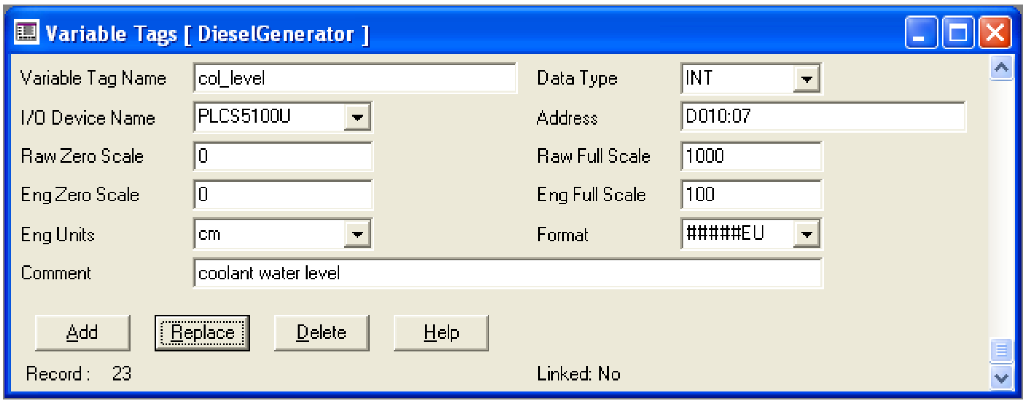
**table (1): process data table**

| (1) Process Variables | | (2) storage in PLC Registers | | | | (3) Citect SCADA Tags | (4) Ranges/Displayed Values | | | | | (5) Displayed AS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DESCRIPTION OF DATA | USED IN | TYPE | UNIT | PLC address Siemens (S5-100U) | I/O | VARIABLES TAG NAME | RAW ZERO | RAW FULL | ENG ZERO | ENG FULL | ENG UNITS | GRAPHICAL BJECT TYPE |
| push button(off line) | control | digital | IODev1 | D010:001.00 | w | system_OFF | | | | | | symbol set |
| push button auto | control | digital | IODev1 | D010:001.01 | w | auto_cmnd | | | | | | symbol set |
| push button manual | control | digital | IODev1 | D010:001.02 | w | manual_cmnd | | | | | | symbol set |
| push button start | control | digital | IODev1 | D010:001.03 | w | start_cmnd | | | | | | symbol set |
| push button stop | control | digital | IODev1 | D010:001.04 | w | stop_cmnd | | | | | | symbol set |
| push button reset | control | digital | IODev1 | D010:001.05 | w | reset_cmnd | | | | | | symbol set |
| push button CB close | control | digital | IODev1 | D010:001.06 | w | CB_close | | | | | | symbol set |
| push button CB open | control | digital | IODev1 | D010:001.07 | w | CB_open | | | | | | symbol set |
| engine indication | monitoring | digital | IODev1 | D010:001.08 | r | en_stat | | | | | | symbol set + event |
| CB 52DGL | monitoring | digital | IODev1 | D010:001.09 | r | CB_stat | | | | | | symbol set + event |
| oil pressure low | control | digital | IODev1 | D010:001.10 | r | press_low | | | | | | symbol set + alarm |
| oil temperature high | control | digital | IODev1 | D010:001.11 | r | oil_temp_H | | | | | | symbol set + alarm |
| over speed | control | digital | IODev1 | D010:001.12 | r | en_speed_high | | | | | | symbol set + alarm |
| coolant level low | control | digital | IODev1 | D010:001.13 | r | col_level_L | | | | | | symbol set + alarm |
| generator over voltage | control | digital | IODev1 | D010:001.14 | r | gen_volt_high | | | | | | symbol set + alarm |
| fuel level low | control | digital | IODev1 | D010:001.15 | r | fuel_level_L | | | | | | symbol set + alarm |
| over load/short circuit | control | digital | IODev1 | D010:002.00 | r | gen_load_high | | | | | | symbol set + alarm |
| fuel level low low | control | digital | IODev1 | D010:002.01 | r | fuel_level_LL | | | | | | symbol set + alarm |
| generator temperature | monitoring | Integer | IODev1 | D010:003 | r | gen_temp | 0 | 1000 | 0 | 100 | deg | Numeric + periodic trend |
| oil temperature | monitoring | Integer | IODev1 | D010:004 | r | Oil_temp | 0 | 1000 | 0 | 100 | deg | Numeric + periodic trend |
| engine speed | monitoring | Integer | IODev1 | D010:005 | r | en_speed | 0 | 10000 | 0 | 3000 | rpm | Numeric + periodic trend |
| fuel level | monitoring | Integer | IODev1 | D010:006 | r | fuel_level | 0 | 1000 | 0 | 200 | cm | Numeric + periodic trend |
| coolant level | monitoring | integer | IODev1 | D010:007 | r | col_level | 0 | 1000 | 0 | 100 | cm | Numeric + periodic trend |




**Figure(9):addressing digital variable**

En_stat is variable tag name in citectHMI/SCADA, engine indication table(1)
Data Type DIGITAL
Address D010:001.08
Comment Digital - Block Number 10: Word Number 1. Bit Number 8
Same as digital variable we addressing integer variable figure(10).




**Figure(10): addressing integer variable**

Col_level is variable tag name in citectHMI/SCADA, coolant level(table1)
Data Type INT.
Address D0010:007.
Comment Data Word - Block Number 10: Word Number 7.

**Step3**:

CitectSCADA comes in two programming languages [6] – Cicode and CitectSCADA VBA(Visual Basic for Application). Cicode is designed specifically for plant monitoring and control application. While CitectSCADA VBA is better suited to interacting with third party objects and applications.With Cicode and CitectSCADA VBA, we have access to and control of all the elements in our runtime system (real time data, historical data, operator displays, alarms, events, reports, trends, security, etc).

These programming languages also give us access to our computer system, including the operating system and communication ports. Cicode is an advanced language that is similar to other high level languages like ‘C’. CitectSCADA VBA is a100% compatible with Microsoft Visual Basic for Application(VBA). CitectSCADA has over 650 SCADA/HMI specific functions included, reducing the need for complex or extensive code, which allows us to quickly and easily design a typically Operator Interface System(OIS) as shown in figure(11).

Menu page will feature the buttons to: display the graphic page representing the field process (diesel generator control) associated with the process data, and display the others pages (trend, alarm, utility, test, report ).

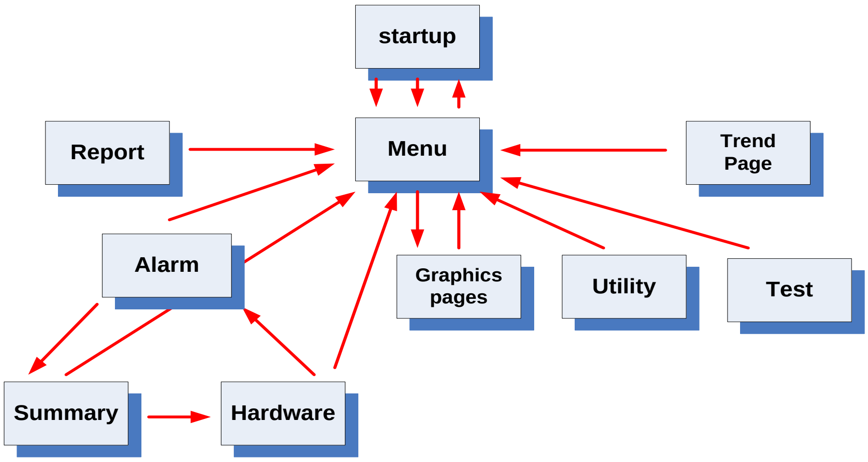
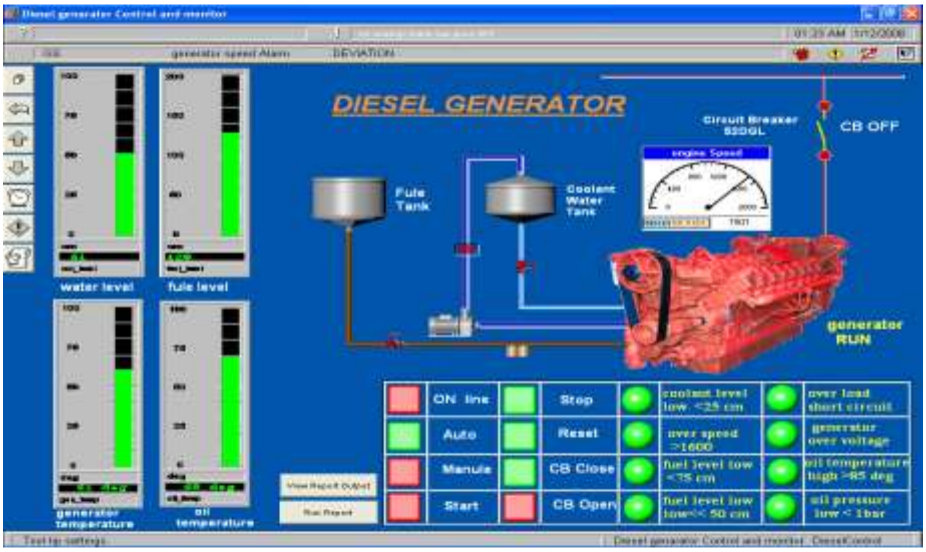



**Figure (11): All designing pages need to create an Operator Interface System(OIS) Using CitectHMI/SCADA.**

### 1.Graphic Page:

Rapid Application Development (RAD) and Graphics of CitectSCADA are based on a simple set of objects, namely rectangles, ellipses, symbol libraries, pipes, and activeX objects. Associated with all these objects is a common set of object properties. These properties allow an object's behavior to be directly linked to our plant variables(table1), as shown in the diesel generator control and monitor page figure(12).




**Figure(12):control and monitor page is the main monitoring page which has the graphic image of plant and most process data**

We create this page by adding the needed objects from citectSCADA library and linking the object properties with the plant variables. For example,

**EX1**-Generator start command: (start_cmnd, I/O(w), as shown in table(1))
Object→property→Input→touch
Toggle(start_cmnd);
End.
**EX2**-Generator monitoring: (en_stat, I/O(r), as shown in table(1))
Object→property→Appearance→general(on/off)
en_stat;
End.

Toggle(sTag) is an implicit function, which changes the digital variable (start_cmnd) every time we touch the object (ex1). As is the same in ex1, ex2 en_stat is a digital variable showing if the engine is running (en_stat=1, ON) or switched off (en_stat=0, OFF).Also, by this property (appearance), we can change the color of the object (red=run, green=stop).

**2. Trend Page:**

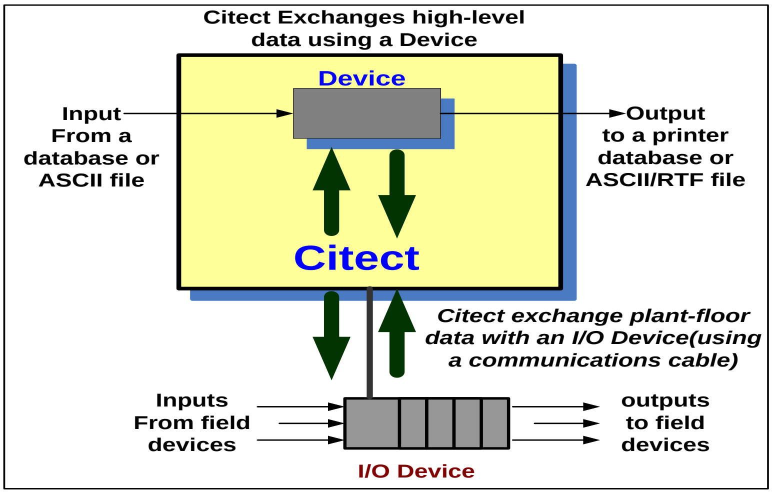
CitectSCADA comes with a host of ready-made trend templates, allowing us to quickly create trend graphs complete with navigation tools and dynamic readouts from the plant floor. We can display trends by linking the citectSCADA variables to the pens of the trend page figure(13). CitectSCADA trends are created from a selection of sample values. Sampling rates can be as low as **10 milliseconds**, and as high as **24 hours**.


**Figure(13):trends page has eight pens, five of which are related to fuel and Coolant level, oil and generator temperature, and engine speed. The sample period of the pens is 1 second.**

**3. Device**

The Other pages, like alarm page and report page, need an associated device.

A Device is a utility that transfers high-level data (such as a report, command log or alarm log) between CitectHMI/SCADA and other elements (such as a printer, database, RTF file , or ASCII file ) in our CitectHMI/SCADA system. as shown in figure(14).




**Figure(14): CitectHMI/SCADA device**

Devices are similar to I/O devices in that they both allow CitectHMI/SCADA to exchange data with other components in the control and monitoring system. We use a device to write data to:

- **RTF** Rich Text Format (formatting, colors and graphics).
- **TXT** Plain ASCII text.
- **DBF** Database file(dbase III).
- **SQL** SQL database (through ODBC-compliant drivers).
- **Printer** Printers(connected to CitectHMI/SCADA computer or network).

**4. Report Page:**

We can request regular reports on the status of the plant and provide information about special conditions in the plant. Reports, like events, can run it periodically or be triggered (or both). They can also be run at any time by using the Cicode function **Report().**

The report format file can contain information such as static text, formatting information Cicode and data from variables.

We define a new device (NAME: DieselLog, FILE NAME:[DATA]:diesel_Rep.rtf , TYPE: ASCII_DEV, NO.FILES: -1, COMMENT: a single report file).

We define an RTF report (NAME :Diesel , REPORT FORMAT FILE :Diesel.rtf , OUTPUT DEVICE: DieselLog).

As we show the report “Diesel” coupled with the device “DieselLog”.

The report “Diesel.rtf” that will be created is shown in figure(15).

We will run the report by using the Function **Report**(“Diesel”), and view it by using the function PageRichTextFile(*hAn* , *Filename* , *nMode* , *nHeight* , *nWidth* ) .

PageRichTextFile ( 35 , "[DATA]:Diesel_rep.rtf", 0 , 600 , 900).

nHeight, nwidth are the dimension of the rich text object in pixels.

nMode is the displaying mod (0: display only(disabled) ).

Filename: device file name.

hAn: The animation point at which to display the rich text object (35).

```
DIESEL GENERATOR REPORT
Time:{TIME(1)      }                    Date:{DATE(2)      }
status of  process variables at the time of this report:
Oil temperature                          {oil_temp:###EU}
Generator speed                          {en_speed:####EU}
coolant level                            {col_level:###EU}
Fuel level                               {fuel_level:###EU}
Generator temperature                    {gen_temp:###EU}
{Cicode}
IF en_stat=1 then
        Print("Diesel generator  RUN") ;
ELSE
        Print("Diesel generator  STOP");
End
IF CB_stat=1 then
        Print("Circuit Breaker 52DGL  CLOSE");
ELSE
        Print("Circuit Breaker 52DGL  OPEN");
End
{End}
---------- Report End -----------
{cicode}
prompt("Report complet");
{End}
```

**Figure(15): Every time we call, a report gives us the value of some process variables like Fuel level, coolant level and if the generator is switched off or running, using Cicode programming language.**

After running the report, the result is shown in figure(16).

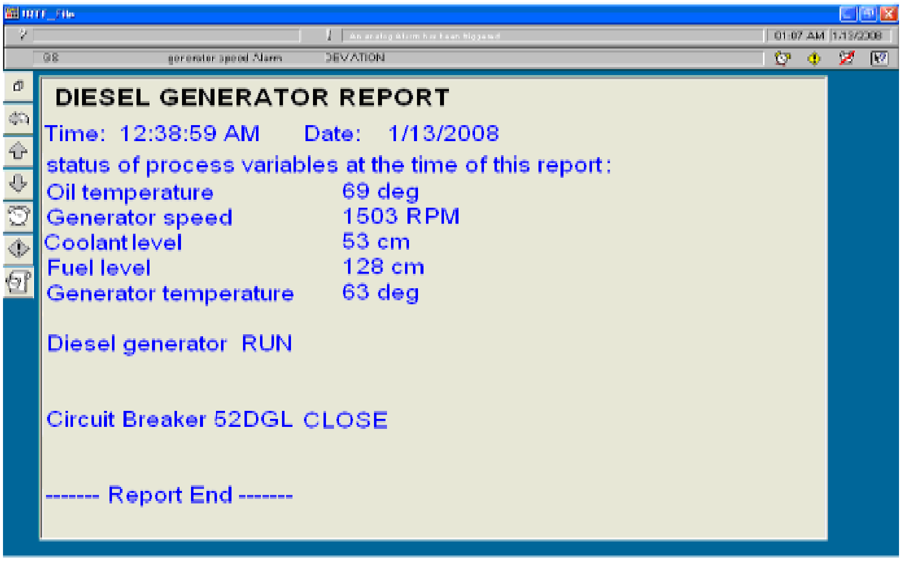



**Figure (16): diesel report: date (1/13/2008) and time (12.38.59 am)**

As reports, alarms need a device to assess the displaying format.

Device(name: AlarmSummary, Format: {Name,16}{Desc,32}{OnTime,11}{Deltatime,11}, Header: [DATA]:DieselAlarmSum.txt, type: dBASE_DEV, No.Files: 7, time: 00:00:00, period: 24:00:00, comment: A daily history file of the Alarm summary).

**5. Alarms Page:**

The CitectSCADA alarm system is fast and reliable, providing us with detailed alarm information in formats that are clear and legible. citectSCADA provides us with many kinds of alarms, which are processed and managed by I/O server; the client can display and acknowledge them. In our case, we use only two kinds: digital alarms and analog alarms.

We will create digital alarms by adding this information to (digital alarm editor):

| Alarm Tag | Alarm Name | Alarm Description | Var Tag A |
|---|---|---|---|
| speed | engine speed high | engine speed >1600 RPM | en_speed_high(table1) |

A similar manner for the analog alarms:

| Alarm Tag | Alarm Name | Var Tag | Setpoint | High High | High | Low | Low Low | Devi-ation | Dead-band | Format |
|---|---|---|---|---|---|---|---|---|---|---|
| CW | Coolant Water Level Alarm | col_level (table1) | 50 | 85 | 75 | 25 | 15 | 5 | 1 | ### |

All kinds of alarms are grouped into two types: Hardware Alarms (equipment such as I/O devices..) and Configured Alarms (which are related to process variables), as shown by the Alarms summary page figure (17).

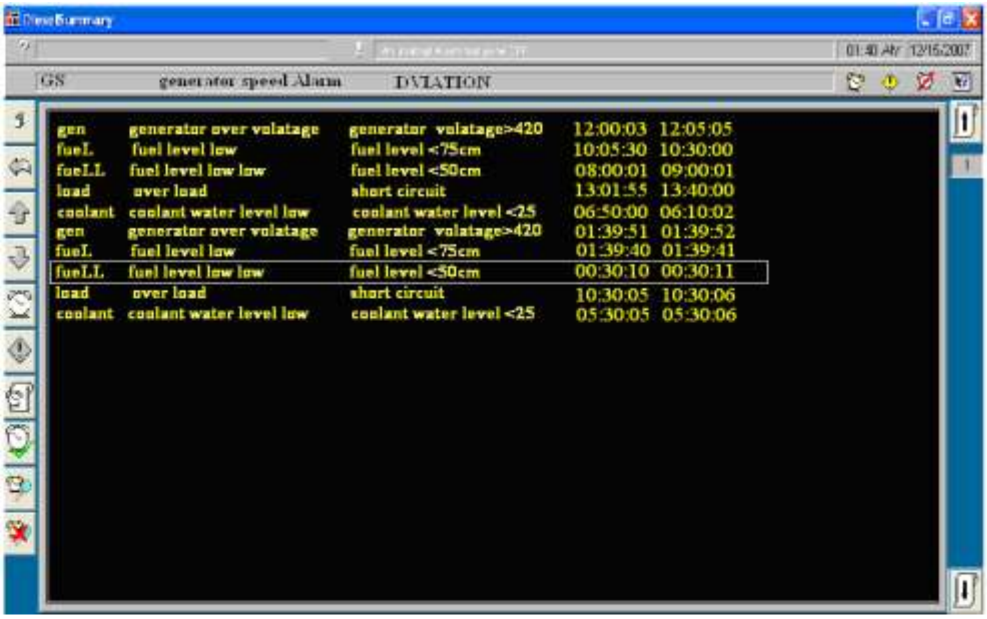



**Figure (17):Alarm summary page shows all abnormal events (alarms) associated with the time and the date of activation and the time of deactivation. In addition, activation alarm is yellow, whereas deactivation alarm is green.**

## Conclusion

In this pape, r we have shown that SCADA system is added to traditional control system of the human interface level. This provides a powerful way in monitoring and making a decision in emergency, which is very important in emergency diesel generators. On the other hand, by the same version of Citect/SCADA, we can build a SCADA system to various diesel generators together. This means SCADA is a flexible and scalable system. Also, it is easy to build SCADA software, and there is no need to write so many code lines to build an Operator Interface System(OIS). In addition, to reduce the cost by counting physical I/O points, SCADA system reduces the labor cost by increasing sensors.

## REFERANCES

1- *SCADA/Model Integration: The Rules for Success* . Bentley Systems, Incorporated. All rights reserved , 2004 .
**http://wateronline.com**/Content/news/article.asp?Bucket=Article&...

2- METSO, AUTOMATION. *Proposed OASYS System for :south florida Municipal water Distinct water management SCADA system.* June 5, 2002.
www.**sfwmd.gov**/gover/wrac/ref_mat/lake_okee_mtg_012506.pdf .

3- ALSTOM T&D EAI / EME . *PLC Preliminary Conception Document*, *File: C001087 / DCP 4MAD0052-C.* SYRIA - BACKUP SUBSTATION METERING SYSTEM (PEEGT). 2003 .

4- KEVIN, S. WHICH CONTROL OPTION DO I GO WITH? DCS, MINI-DCS OR PLC? . 4507 Pacific Hwy. E., Ste A . Tacoma , WA 98424 .

5- CONEX ; MITSUBISHI HEAVY INDUSTRIAL ."*Emergency Diesel Generator PLC" Logic diagram* , Plant Auxiliaries/Instruction Manual VOL.E-VIII . Public Establishment of Electricity for Generation and Transmission
http//www.ncsa.uiuc.edu/WebSGML/ WebSGML.sgml
(PEEGT). AL-ZARA THERMAL POWER STATION . 2002 .

6- AUSTRALIA PR10465 .*CitectSCADA Technical Overview* . http://www.Citect.com

7- SIEMENS. *simatic S5: S5-100U programmable controller/ system Manual CPU 100/ 102/103* .4th .ed .EWA 4NEB 812 6120-02b. http//www.siemens.com .

8-LAROSE, C. J. ; CLAVETTE , C. J . *history in the making an intimate look at data Collection and reporting* . CDM one glen lakes 8140 walnut hill lane , suite 1000 Dallas , Texas 75231 .

9-ScanData . *telemetry & SCADA Handbook* . , *RTU and PLC communication* . , *Distributing SCADA Data over DDE, SQL and on the WEB* .
http://www.scan_data.com . 2004.

10-CLARK,G.; REYNDERS, D.; WRIGHT, E .*practical modern SCADA protocols*.1st. ed. , Elsevier & Newnes , Great Britain , 2004 , 537.

11-DANEELS, A.;SALTER, W. *What is SCADA?.* International conference on Accelerator and Large Experimental Physics Control System ,1999,Trieste,Italy.

12-ALEINIKOV, V. ; DREYZINA,G. ; ALEINIKOV, A. *Custom HMI for Commercial SCADA.* DUBNA, Russian Federation.
www.**lnf.infn.it**/conference/pcapac2002/TALK/WE-P01/WE-P01

13- GADBOIS, M . R. *indusoft web studio builds HMI/SCADA applications both large and small.* InduSoft (Tcole for Automation) . Austin,Texas-February 22, 2004 .
http://www.IonduSoft.com .

14-PARR , E.A.; MSC ;CENG; MIEE ; MINSTMC . *programmable controllers an engineer's guide.* 3rd . ed ., Newnes, Great Britain, 2003, 429.

15- GALLINA . L . *AS511 Protocol Notes* .last revised: March 17, 2004 .
http://www.runmode.com .

16- SIEMENS . *simatic HMI communication / user's manual* . release 05/99 . 6AV3991- 1BC05-1AB0 . , Impressum ,1999 , http//www.siemens.com .